

## **General Disclaimer**

### **One or more of the Following Statements may affect this Document**

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

JAN 21 1981

EW-L1-00709  
JSC-16855

# AgRISTARS

"Made available under NASA sponsorship  
in the interest of early and wide dis-  
semination of Earth Resources Survey  
program information and without liability  
for any use made thereof."

A Joint Program for  
Agriculture and  
Resources Inventory  
Surveys Through  
Aerospace  
Remote Sensing

January 1981

## Early Warning and Crop Condition Assessment

COMPUTER PROGRAM DOCUMENTATION FOR THE  
PATCH SUBSAMPLING PROCESSOR

E82-10099  
NASA CR-161040

M. J. Nieves, S. O. O'Brien, and J. K. Oney

(E82-10099) COMPUTER PROGRAM DOCUMENTATION  
FOR THE PATCH SUBSAMPLING PROCESSOR  
(Lockheed Engineering and Management) 210 p  
HC A10/ME A01 CSCL 02C

N82-22541

Unclass  
G3/43 00093

Lockheed Engineering and Management Services Company, Inc.  
1830 NASA Road 1, Houston, Texas 77058



NASA



Lyndon B. Johnson Space Center  
Houston, Texas 77058

1. Report No. EW-L1-00709; JSC-16855		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Computer Program Documentation for the Patch Subsampling Processor				5. Report Date January 1981	
				6. Performing Organization Code	
7. Author(s) M. J. Nieves, S. O. O'Brien, J. K. Oney Lockheed Engineering and Management Services Company, Inc.				8. Performing Organization Report No. LEMSCO-15119	
				10. Work Unit No.	
9. Performing Organization Name and Address Lockheed Engineering and Management Services Company, Inc. 1830 NASA Road 1 Houston, Texas 77058				11. Contract or Grant No. NAS 9 15800	
				13. Type of Report and Period Covered Procedures Document	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 Technical Monitor: J. D. Erickson				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract  This report documents the software which has been developed in response to Action Document 63-21-37-3343-01, "Sampling Scheme for Early Warning Alarm." The total software system referred to as the Patch/Skip Sampling System is described in detail.					
17. Key Words (Suggested by Author(s))  IMDACS Patch Skip				18. Distribution Statement	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 210	
				22. Price*	

COMPUTER PROGRAM DOCUMENTATION FOR THE PATCH SUBSAMPLING PROCESSOR

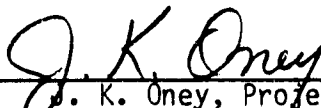
Job Order 73-366

This report describes Spectral Analysis activities of the  
Early Warning/Crop Condition Assessment project.

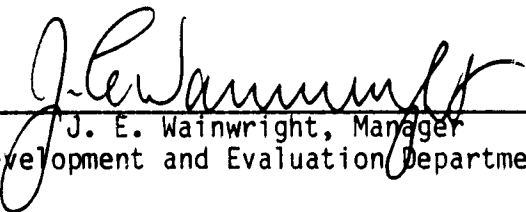
PREPARED BY

M. J. Nieves, S. O. O'Brien, and J. K. Oney

APPROVED BY



J. K. Oney, Project Manager  
Early Warning Project Office



J. E. Wainwright, Manager  
Development and Evaluation Department

LOCKHEED ENGINEERING AND MANAGEMENT SERVICES COMPANY, INC.

Under Contract NAS 9-15800

For

Earth Observations Division  
Space and Life Sciences Directorate  
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION  
LYNDON B. JOHNSON SPACE CENTER  
HOUSTON, TEXAS

January 1981

LEMSCO-15119



PRECEDING PAGE BLANK. NOT FILMED

CONTENTS

Section	Page
1. SCOPE.....	1-1
2. SYSTEM OVERVIEW.....	2-1
3. SUBSYSTEM 1 - READTP.....	3-1
3.1 <u>UNIT 1 - HDREAD</u> .....	3-5
3.1.1 UNIT 1, LEVEL 2 - HDCODE.....	3-8
3.1.1.1 <u>Unit 1 Level 3 - FRONT</u> .....	3-10
3.1.1.2 <u>Unit 1, Level 3 - ETAATE</u> .....	3-11
3.1.1.3 <u>UNIT 1, LEVEL 3 - B2I</u> .....	3-12
3.1.1.4 <u>Unit 1, Level - A2I</u> .....	3-13
3.1.1.5 <u>Unit 1, Level 3 - LATLON</u> .....	3-14
3.1.1.6 <u>Unit 1, Level 3 - JULIAN</u> .....	3-15
3.1.2 UNIT 1, LEVEL 2 - HPROS.....	3-16
3.1.3 UNIT 1, LEVEL 2 - IBYTE.....	3-17
3.2 <u>UNIT 2 - INTOCT</u> .....	3-18
3.3 <u>UNIT 3 - HDRFLN</u> .....	3-19
3.3.1 UNIT 3, LEVEL 2 - UNVRSL.....	3-20
3.4 <u>UNIT 4 - HDT</u> .....	3-22
3.4.1 UNIT 4, LEVEL 2 - LBDORDR.....	3-23
3.4.2 UNIT 4, LEVEL 2 - PBORDR.....	3-24
3.4.3 UNIT 4, LEVEL 2 - LCODE.....	3-25
3.4.4 UNIT 4, LEVEL 2 - PARTP.....	3-27
3.4.5 UNIT 4, LEVEL 2 - HIST.....	3-29
3.5 <u>UNIT 5 - LANDSAT</u> .....	3-30

Section	Page
3.6 <u>UNIT 6 - HDRHIS</u> .....	3-31
4. SUBSYSTEM 2 - SKIP.....	4-1
4.1 <u>UNIT 1, LEVEL 1 - SKPRD</u> .....	4-3
4.1.2 UNIT 1, LEVEL 2 SKPWRT.....	4-5
4.1.2.1 <u>Unit 1, Level 3 - SKPHST</u> .....	4-7
5. SUBSYSTEM 3 - PATCHP.....	5-1
5.1 <u>UNIT 1, LEVEL 1 - HEADER</u> .....	5-5
5.2 <u>UNIT 1, LEVEL 1 - PATCHRD</u> .....	5-6
5.3 <u>UNIT 1, LEVEL 1 - GAMMA</u> .....	5-7
5.3.1 UNIT 1, LEVEL 2 - CALIB.....	5-8
5.3.2 UNIT 1, LEVEL 2 - MEAN.....	5-10
5.3.3 UNIT 1, LEVEL 2 - SCRALL.....	5-12
5.4 <u>UNIT 1, LEVEL 1 - HAZCOR</u> .....	5-13
5.4.1 UNIT 1, LEVEL 2 - KAUTH.....	5-14
5.5 <u>UNIT 1, LEVEL 1 - VIN</u> .....	5-15
5.6 <u>UNIT 1, LEVEL 1 - PATRPT</u> .....	5-17
6. POSPP.....	6-1
7. REFERENCES.....	7-1
Appendix	
A. LISTING OF PATCH PROCESSOR.....	A-1
B. LISTING OF SKIP PROCESSOR.....	B-1
C. LISTING OF PATCH P PROCESSOR.....	C-1
D. LISTING OF POST P PROCESSOR.....	D-1

## FIGURES

Figure		Page
2-1	Flow diagram of Subsystem 1.....	2-2
2-2	Flow diagram of Subsystem 2.....	2-3
2-3	Flow diagram of Subsystem 3.....	2-4

## ABBREVIATIONS

ASCII	The ASCII Character Set
AVI	Ashburn Vegetative index
CCT	Computer Compatible Tape
DVI	Difference Vegetative Index
EBCDIC	The EBCDIC Character Set
GIN5	Green Index Number 5
GIN15	Green Index Number 15
GIN30	Green Index Number 30
GREEN	Green Number
HDT	High Density Tape
IMDACS	Integrated Multivariate Data Analysis and Classification System
LAI	Leaf Area Index
LARSYS	Laboratory for the Application of Remote Sensing System
LEC	Lockheed Electronics Company
PVI	Perpendicular Vegetative Index
RVI	Red Vegetative Index
SPU	Scene Processing Unit
TVI	Transformed Vegetative Index
X-STAR	Haze Correction Algorithm

## 1. SCOPE

This report documents the software which has been developed in response to Action Document 63-21-37-3343-01, "Sampling Scheme for Early Warning Alarm." The total software system referred to as the Patch/Skip Sampling System is described in detail.

## 2. SYSTEM OVERVIEW

The programs contained in this document are intended to provide a way to extract a sample from a full-frame scene and summarize it in a useful way. The sample in each case was chosen to fill a 512-by-512 pixel (sample-by-line) image since this is the largest image that can be displayed on the Integrated Multivariant Data Analysis and Classification System (IMDACS) system. This sample size provides one megabyte of data for manipulation and storage and contains about 3 percent of the full-frame data.

Two extraction strategies (methods) are represented in the programs. In the skip method, every sixth pixel from every sixth line is taken to show a sampled image which preserves the gross structure of the full-frame image. In the patch method, 32-sample by 32-line pixel squares (patches) are taken from the full-frame image on a 16-by-16 square grid. These 256 squares of data can provide a summary of the full frame while retaining the full resolution field structure for analyst interpretation.

In both cases, the 512-by-512 pixel image is further summarized by the patch image processor. This processor computes means for 256 32-by-32 pixel squares which constitute the 512-by-512 pixel image. For each square, the screening algorithm is run and the channel means are computed along with a screening summary of the rejected pixels. The mean vegetation index, truncated at 0, is computed for values GREEN, AVI, PVI, TVI7, and LAI. The values of GIN5, GIN15, and GIN30 are also computed. These may be computed with or without X-STAR haze correction (ref. 1).

Thus, 256 measurements for each of 8 vegetation indexes are available over a 100-mile (161-kilometer) square, computed in 2 ways.

With these data from several segments, the vegetation indexes and the two modes of sampling the full-frame image are compared.

Flow diagrams of Subsystems 1, 2, and 3, are presented in figures 2-1, 2-2, and 2-3.

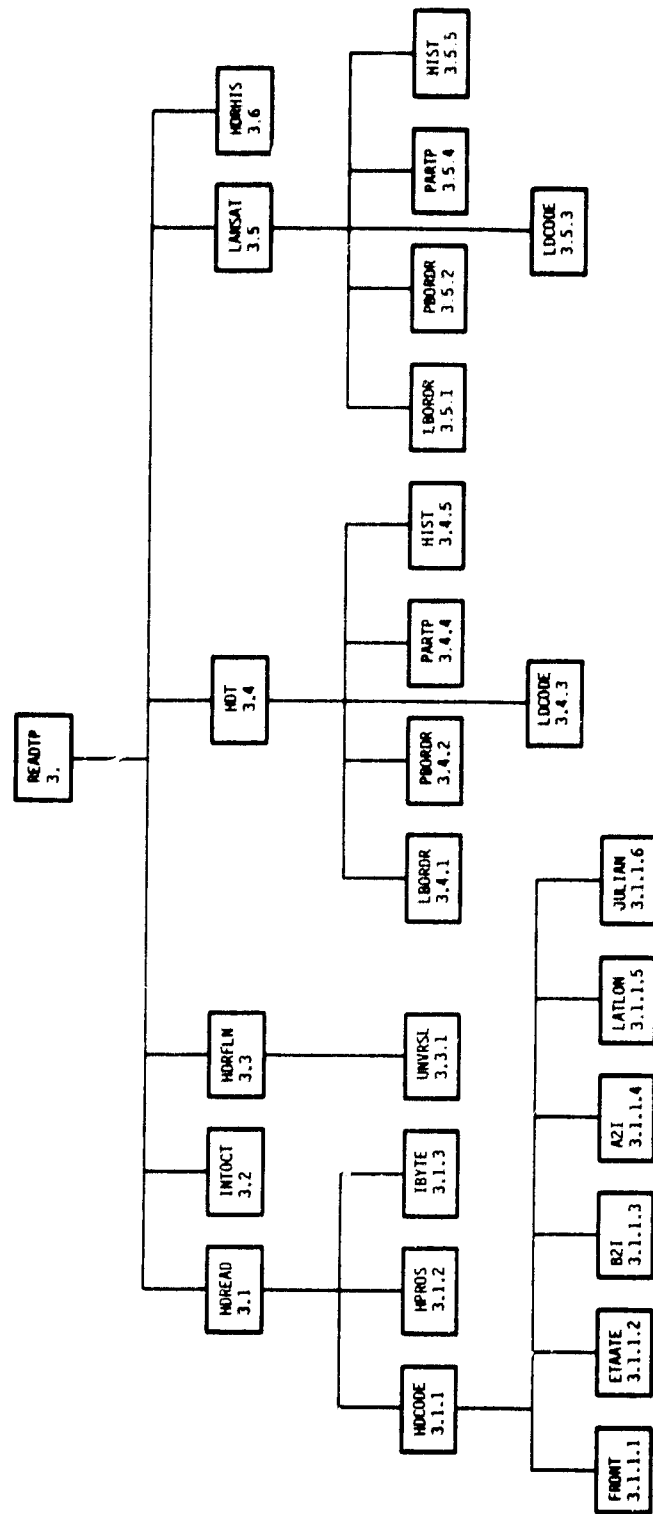


Figure 2-1.- Flow diagram of Subsystem 1.

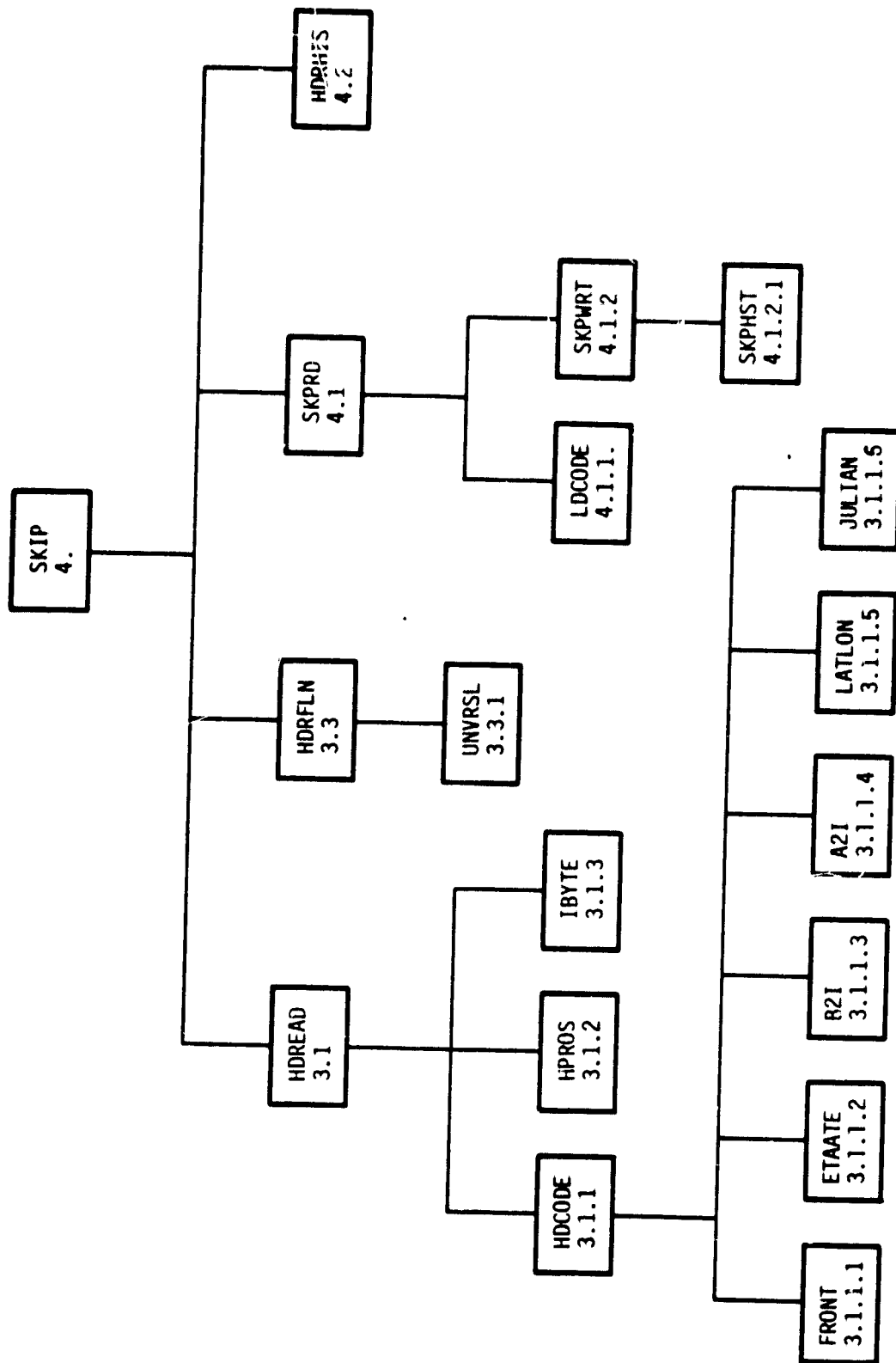


Figure 2-2.- Flow diagram of Subsystem 2.



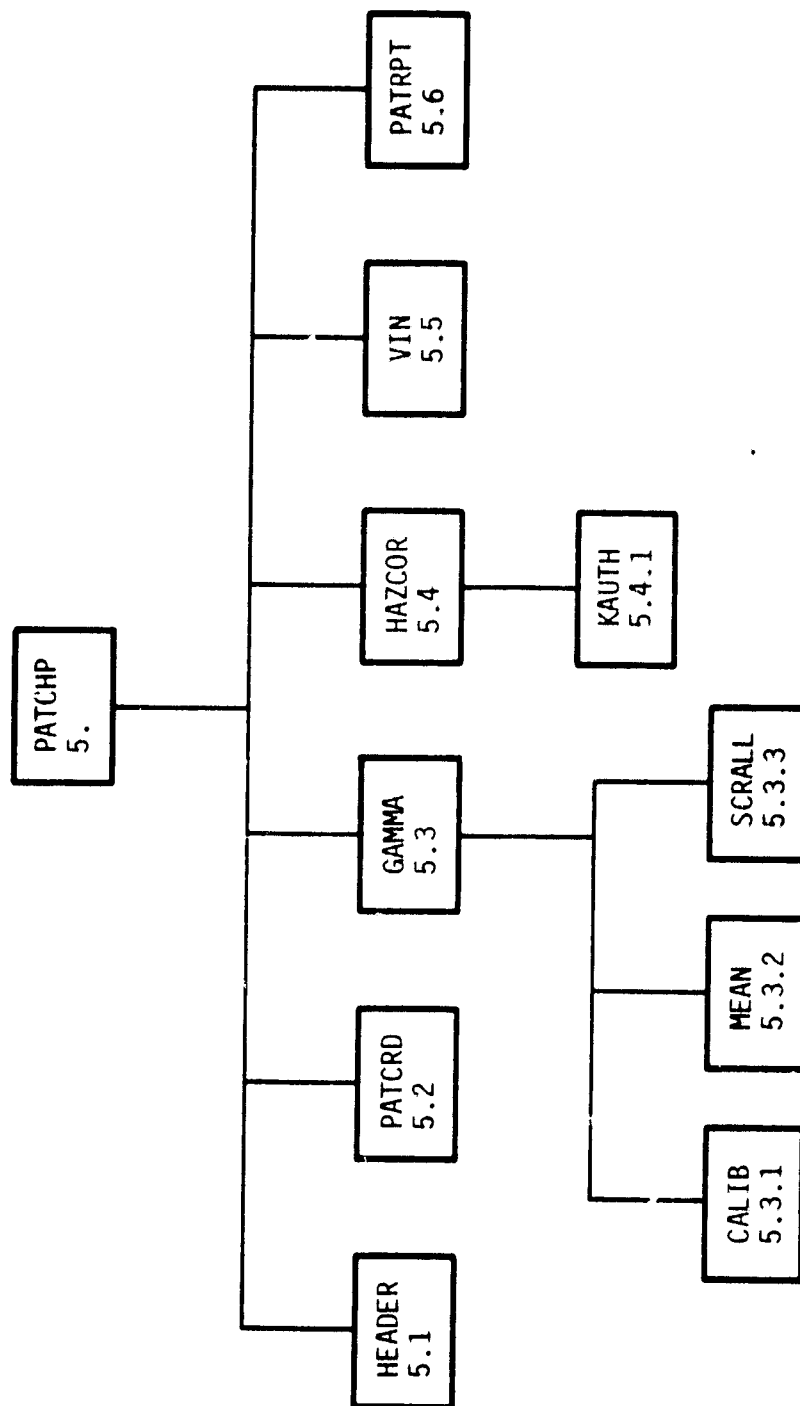


Figure 2-3.- Flow diagram of Subsystem 3.

### 3. SUBSYSTEM 1 - READTP

The READTP program is the main routine of the patch image processor. It is the driver routine for reading Landsat 1-, Landsat 2-, or Universal-formatted tape. In addition, it is the driver routine for creating the patch image in the IMDACS form.

CALLING SEQUENCE: PROGRAM READTP

CALLING ARGUMENTS: Not applicable.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record format</u>
RPARAM.DAT	User interface	(17X,I5,/,17X,I5,/12X,I3,/,10X,6A1)
???????.IMH	IMDACS header file	Refer to IMDACS documentation, volumes 1 and 2
???????.IMD	IMDACS data file	Refer to IMDACS documentation, volumes 1 and 2

COMMON BLOCKS: The following abbreviations are used in tables throughout this document.

A = alphanumeric  
I = integer  
I/O = input output  
L = logical  
R = real

Common blocks for the READTP routine are /COMH/, /LABEL/, and /HIST/.

/COMH/COMMON BLOCK

<u>Name</u>	<u>Relative word number</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
ST	1	I/O	I	Starting pixel number on computer compatible tape (CCT).
SE	2	I/O	I	Ending pixel number on CCT.
LS	3	I/O	I	Line start number on full frame.
LE	4	I/O	I	Line end number on full frame.
NOCHAN	5	I/O	I	Number of channels on tape.
NOSAMP	6	I/O	I	Number of pixels per scan line per channel.
SCNID	7-12	Header	A	Scene identification.
SUNEL	13	Header	I	Sun elevation angle.
SUNAZ	14	Header	I	Sun azimuth angle.
SNSHD	15	Header	I	Sensor heading.
NREV	16	Header	I	Orbit revolution number.
EXYR	17	Header	I	Exposure year.
EXDAY	18	Header	I	Exposure day (Julian).
DAY	19	Header	I	Scene identification day.
HR	20	Header	I	Scene identification hour.
MIN	21	Header	I	Scene identification minute.
SEC	22	Header	I	Scene identification second.
SEQNO	23-24	Header	I	CCT sequence number (N of M).
FLAT	25-26	Header	R	Center of the full frame (latitude).
FLONG	27-28	Header	R	Center of the full frame (longitude).
NLAT	29-30	Header	R	NADIR (latitude).
NLONG	31-32	Header	R	NADIR (longitude).

/LABEL/Common Block

<u>Name</u>	<u>Relative word number</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
NAME	1-3	Error	A	Name of routine.
MISNO	4	Header	I	Mission number.
E	5	Header	I	Mission number.

/HIST/Common Block

<u>Name</u>	<u>Relative word number</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
H4	512	Array	R	Histogram for band 4.
H5	1025	Array	R	Histogram for band 5.
H6	1537	Array	R	Histogram for band 6.
H7	2049	Array	R	Histogram for band 7.

SUBROUTINES CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
HDREAD	Header record read.	3.1
INTOCT	Integer to OCTAL.	3.2
HDRFLN	Header file.	3.3
HDT	High density full-frame format.	3.4
LANDSAT	Landsat (1 and 2) format.	3.5
HDRHIS	Header histogram.	3.6

CALLED BY: Not applicable.

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
BFSZ	Buffer size in bytes.
BUF	Buffer for image data.
DELTAL	Change in center point line.
DELTAP	Change in center point pixel.
IE	Error flag.
NOML	Nominal center point line.
NOMP	Nominal center point pixel.
NPOINT	Array containing nominal center point in degrees.
STARTL	Line in which extracting should start.
STARTP	First pixel to be extracted during patch extraction from full frame.
STOPP	Last pixel to be extracted during patch extraction from full frame.

### 3.1 UNIT 1 - HDREAD

The HDREAD subroutine reads the header data records from a foreign imagery tape in one of the following formats: Universal, Landsat 1 or 2, or LARSYS II or III.

CALLING SEQUENCE: HDREAD(LUN,IFRM,HEADER,BFSZ,EOF,PC,IE,BUFFER,INS,HDRADR,ISS)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN	I/O	I	Logical unit number assigned to the tape.
IFRM	Process	I	Format type of the image tape.
HEADER	Array	I	Buffer.
BFSZ	Process	I	Buffer size in bytes.
EOF	Process	I	Flag which, if=1, indicates end of file.
PC	Process	I	Parity counter.
IE	Process	I	Error code.
BUFFER	Array	I	Buffer for data.
INS	Process	I	Number of scan lines.
HDRADR	Process	I	Address of the array for WTQIO.
ISS	Process	I	Starting scan line.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record format</u>
TAPE	Input	Refer to Landsat or Universal formats.

COMMON BLOCKS: The common block for subroutine HDREAD is /HCOM/.

/HCOM/COMMON BLOCK

<u>Name</u>	<u>Relative word number</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
SS	1	Process	I	Sample start.
SE	2	Process	I	Sample end.
LS	3	Process	I	Line start.
LE	4	Process	I	Line end.
NRPDS	5	Process	I	Number of records per data set (data set is a scan line).
NDSPR	6	Process	I	Number of data sets per record.
NCPR	7	Process	I	Number of channels per record.
NRPC	8	Process	I	Number of records per channel.
ANCL	9	Process	I	Length of ancillary block.
NC	10	Process	I	Number of channels in the image.
NS	11	Process	I	Number of samples per channel per scan line.
NBIT	12	Process	I	Number of bits in a pixel.
DOI	13	Process	I	Data order indicator.
NCAR	14	Process	I	Same as NCPR.
SVD	15	Process	I	Start of the video within a data set.
RSIZ	16	Process	I	Tape record size.
PSKIP	17	Process	I	Pixel offset for reading data from tape.
HSIZ	18	Process	I	Number of words in the header OAT.
CALP	19	Process	I	Used in calculating offset bytes in data.
CERR	20	Process	I	Catastrophic error. Calling program should terminate if error $\neq$ 0.
NSPR	21	Process	I	Number of pixels per record.

SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
HPROC	Header process.	3.1.2
HDCODE	Header decode.	3.1.1

CALLED BY: The HDREAD subroutine is called by the driver routine READTP (see section 3).

LOCAL VARIABLES: Local variables for subroutine HDREAD are listed in common block /HCOM/ of the HDREAD subroutine in this section.



### 3.1.1 UNIT 1, LEVEL 2 - HDCODE

The HDCODE subroutine decodes selected portions of tape header records in one of the following formats: Universal, Landsat 1 and 2, or LARSYS II and III.

CALLING SEQUENCE: SUBROUTINE HDCODE(IF,H,IE,LVL)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
IF	Process	I	Variable indicating format of header; if = 1, indicates Universal, if = 2, indicates Landsat, if = 3, indicates LARSYS.
H	Array	A	Header record exactly as read from tape.
IE	Process	I	Variable indicating error code; if = 0, no error; if = 1, not 1, 2, or 3.
LVL	Process	I	Variable which indicates Landsat level indicator; if = 1, level 1, if = 2, level 3.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record Format</u>
Tape header	Input	Landsat format; refer to the Landsat documentation.

COMMON BLOCKS: Common blocks for the HDCODE subroutine are /COMH/, /LATON/, and /LABEL/. For descriptions of common blocks /COMH/ and /LABEL/, refer to program READTP (section 3).

#### /LATLON/COMMON BLOCK

<u>Name</u>	<u>Relative word number</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LATD	1	Process	A	Latitude direction.
LONGD	2	Process	A	Longitude direction.

SUBROUTINES CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
FRONT	Delete blanks from string.	3.1.1.1
ETAATE	EBCDIC to ASCII, ASCII to EBCDIC.	3.1.1.2
B2I	Binary to integer.	3.1.1.3
A2I	Alpha to integer.	3.1.1.4
LATLON	Converts from degrees and minutes to degrees and tenths of a degree.	3.1.1.5
JULIAN	Converts Gregorian calendar to Julian calendar.	3.1.1.6

CALLED BY: The HDCODE routine is called by subroutine HDREAD (see section 3.1).

LOCAL VARIABLES: Local variables for subroutine HDCODE are listed in common block /COMH/. Refer to the driver routine READTP, section 3.

#### 3.1.1.1 Unit 1, Level 3 - FRONT

The FRONT subroutine left justifies an ASCII character string, removes any leading blanks, and blank fills the array after the last character is shifted.

CALLING SEQUENCE: SUBROUTINE FRONT(I,N)

<u>Var</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
I	array	A	String which will be examined for blanks.
N	Process	I	The length of the string in characters.

FILE: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: Subroutine FRONT is called by the subroutine HDCODE (see section 3.1.1).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
K	Counter for the number of characters examined.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

### 3.1.1.2 Unit 1, Level 3 - ETAATE

The ETAATE subroutine converts an EBCDIC character to an ASCII character and an ASCII character to an EBCDIC character one byte at a time.

CALLING SEQUENCE: SUBROUTINE ETAB(H,ARG,NCHAR)

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
H	Array	A	String input of EBCDIC character; may be one character.
ARG	Array	A	String output of ASCII character; may be one character.
NCHAR	Process	I	Number of characters to be converted.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The ETAATE subroutine is called by subroutine HDCODE (see section 3.1.1).

LOCAL VARIABLES: Not applicable.

### 3.1.1.3 Unit 1, Level 3 - B2I

The B2I subroutine converts binary data to integers.

CALLING SEQUENCE: Function B2I(H,B1,B2)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
H	Array	binary	Array containing the string on which to operate.
B1	Process	I	First character in string on which to operate.
B2	Process	I	Last character in string on which to operate.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The B2I subroutine is called by subroutine HDCODE (see section 3.1.1).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
C	The scratch pad word containing the byte on which to operate.

#### 3.1.1.4 Unit 1, Level 3 - A2I

The A2I subroutine converts ASCII data to integer data.

CALLING SEQUENCE: Function A2I(H,B1,B2)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
H	Array	Binary	Array containing the string on which to operate.
B1	Process	I	First character on which to operate in string.
B2	Process	I	Last character on which to operate in string.

FILE: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINE CALLED: Not applicable.

CALLED BY: The A2I subroutine is called by subroutine HDCODE (see section 3.1.1).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
C	The scratch pad word containing the byte on which to operate.

### 3.1.1.5 Unit 1, Level 3 - LATLON

The LATLON subroutine converts the latitude and longitude from degrees and minutes to degrees and tenths of a degree.

CALLING SEQUENCE: REAL FUNCTION LATLON(H,B1,B2)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
H	Array	Byte	Array containing the latitude and longitude.
B1	Process	I	First character in the array H desired.
B2	Process	I	Last character in the array H desired.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The LATLON subroutine is called by the HDCODE subroutine (see section 3.1.1).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
LM	Decimal that places calculation for numeric packing.
LL	Decimal that places calculation for numeric packing.

#### 3.1.1.6 Unit 1, Level 3 - JULIAN

The JULIAN subroutine converts the Gregorian calendar date to the Julian calendar integer.

CALLING SEQUENCE: FUNCTION JULIAN(M,D,Y)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
M	Process	I	Month value.
D	Process	I	Day value.
Y	Process	I	Year value.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINE CALLED: Not applicable.

CALLED BY: The JULIAN subroutine is called by the HDCODE subroutine (see section 3.1.1).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
Julian	The Julian date in integer form.



### 3.1.2 UNIT 1, LEVEL 2 - HPROS

The HPROS subroutine builds a common block of header data for subsequent program use.

CALLING SEQUENCE: HPROS(LUN1,HEADER,FMT,EOF,PRTY,IPRM)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN1	Input	I	Logical unit number assigned to the tape input.
HEADER	Array	A,I	Buffer array used to store the header. Data from the tape header.
FMT	Process	I	Format number of the headers, 1 = Universal, 2 = Landsat, 3 = LARSYS.
EOF	Flag	I	End-of-file indicator; 1 = EOF.
PRTY	Process	I	Parity count.
IPRM	Process	I	Parameter value.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record Format</u>
Tape	Input	Input data tape.

COMMON BLOCKS: The common block for the HPROS subroutine is /HCOM/. Refer to subroutine HDREAD (section 3.1) for a description of this common block.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The HPROS subroutine is called by subroutine HDREAD (see section 3.1).

LOCAL VARIABLES: Local variables for subroutine HPROS are listed in common block /HCOM/. Refer to the HDREAD subroutine, section 3.1.

### 3.1.3 UNIT 1, LEVEL 2 - IBYTE

The IBYTE subroutine converts integers to byte form, one byte at a time.

CALLING SEQUENCE: FUNCTION IBYTE(START,STRING)

CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
START	Process	I	Starting position on string.
STRING	Array	I	Array containing the values which require change.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINE CALLED: Not applicable.

CALLED BY: The IBYTE subroutine is called by subroutine HDREAD (see section 3.1).

LOCAL VARIABLES: Not applicable.

### 3.2 UNIT 2 - INTOCT

The INTOCT subroutine converts the Julian date to OCTAL and converts the version number to OCTAL.

CALLING SEQUENCE: Subroutine INTOCT(INTEGR,OCTAL)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
INTEGR	Process	I	Integer to be converted to OCTAL.
OCTAL	Process	A	Integer value following conversion to OCTAL.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The INTOCT subroutine is called by the driver routine READTP (see section 3).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
VAL	Temporary storage word for calculation.
X	Remainder of a division.
Z	Counter for the number of iterations.

### 3.3 UNIT 3 - HDRFLN

The HDRFLN subroutine writes a header record in IMDACS consisting of a file header record and a Universal imagery header record (SPU format header). The file header and the format header record consist of 1536 words each.

CALLING SEQUENCE: SUBROUTINE HDRFLN(LUN2,FLNIMH,UIFHR,SUNEL,FLAT,FLONG)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN2	In/Out	I	Logical unit number assigned to a file on the disk.
FLNIMH	Array	A	Array containing the file name.
UIFHR	Array	A,I	Array containing the header data.
SUNEL	Process	I	Sun elevation angle.
FLAT	Process	R	Format center latitude.
FLONG	Process	R	Format center longitude.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record Format</u>
???????.IMH	Output	Header file in IMDACS format.

COMMON BLOCKS: Not applicable.

#### SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
UNVRSL	Universal	3.3.1

CALLED BY: The HDRFLN subroutine is called by the driver routine READTP (see section 3).

LOCAL VARIABLES: Not applicable.

### 3.3.1 UNIT 3, LEVEL 2 - UNVRSL

The UNVRSL subroutine builds an SPU universal header record for imagery files.

CALLING SEQUENCE: SUBROUTINE UNVRSL(LUN2,SUNEL,FLAT,FLONG)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN2	I/O	I	Logical unit number assigned to a file on the disk.
SUNEL	Process	I	Sun elevation angle.
FLAT	Process	R	Frame center latitude.
FLONG	Process	R	Frame center longitude.

FILES: Not applicable.

COMMON BLOCKS: Common block for the UNVRSL routine are /LABEL/ and /LATLON/. For a description of /LABEL/, refer to the driver routine READTP (section 3). For a description of /LATLON/, refer to subroutine HDCODE (section 3.1.1).

SUBROUTINES CALLED: Not applicable.

CALLED BY: The UNVRSL subroutine is called by subroutine HDRFLN (see section 3.3).

LOCAL VARIABLES: Local variables for the UNVRSL routine are listed under variables in the IMDACS documentation for the header file.

### 3.4 UNIT 4 - HDT

The HDT subroutine extracts patch images from the full-frame image, with processing taking place one strip of CCT at a time. It further builds a 16-by-16 patch frame image consisting of 32-sample by 32-line pixel patches. Each pixel in a patch is composed of bands 4 through 7. This subroutine uses HDT full-frame images.

CALLING SEQUENCE: Subroutine HDT(LUN1,LUN2,BUF,BFS2,FLNIMD,EOF,IE,BUFFER)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN1	In/out	I	Logical unit number assigned to the tape.
LUN2	In/out	I	Logical unit number assigned to the disk.
BUF	Array	I	Array containing the data.
BFSZ	Array	I	Size of the data array.
FLNIMD	Array	I	Array containing the name of the output file.
EOF	Array	I	End of file.
IE	Array	I	Error code.
BUFFER	Array	I,A	Used as a scratch pad for pixel and header data.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record Format</u>
?????.IMD	Output file	Consult with the IMDACS system documentation.

COMMON BLOCKS: The common blocks for the HDT routine are /COMH/ and /LABEL/. For a description of each, refer to routine READTP (section 3).

SUBROUTINES CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
LBORDR	Line border	3.4.1
PBORDR	Pixel border	3.4.2
LDCODE	Line decode	3.4.3
HIST	Histogram	3.4.5

CALLED BY: The HDT subroutine is called by the driver routine READTP (see section 3).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
BLOCKN	Block number.
ENDCCT	Parameter for reading to the end of the file.
LCOUNT	Line counter for the patch image.
LEND	Ending line within the full frame.
LSTART	Starting line within the full frame.
PATCH	Current patch number.
PIXELS	Pixel location in PIXBUF.
PTCH	Flag indicating that the patch counter should be decremented.
SAVEP	Last patch processed on previous CCT.
STARTP	Starting pixel location in patch extraction.
STOPP	Last pixel location in patch extraction has been readjusted.

### 3.4.1 UNIT 4, LEVEL 2 - LBORDR

The LBORDR subroutine determines when imagery data desired from full frame is not available; patch lines are outside of the full frame.

CALLING SEQUENCE: SUBROUTINE LBORDR(STARTL,NEXTL,LS,LE,LEND,LSTART)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
STARTL	Process	I	Calculated starting line from calling routine.
NEXTL	Flag	L	Flag which causes the line not to be processed.
LS	Process	I	Starting line of full frame.
LE	Process	I	Last line of the full frame.
LEND	Process	I	Last line within the full frame.
LSTART	Process	I	First line within the full frame.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The LBORDER subroutine is called by subroutines HDT and LANSAT (see sections 3.4 and 3.5, respectively).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
LEND	Line end.
LSTART	Line start.
NEXTL	Flag which, if = TRUE, instructs program to process next full frame scan line.



### 3.4.2 UNIT 4, LEVEL 2 - PBORDR

The PBORDR subroutine determines when the imagery data desired from the full frame is not available; patch pixels are outside of the full frame.

CALLING SEQUENCE: SUBROUTINE PBORDR(STARTP,STOPP,STOP,NEXTP,SE,SAT)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
STARTP	Process	I	Starting pixel number provided by the calling routine; altered if necessary for each patch.
STOPP	Process	I	Stop pixel for each patch; altered portion of patch is on the next CCT.
STOP	Process	I	Stop pixel for each patch.
NEXTP	Flag	L	Flag indicating if the patch should be processed.
SE	Process	I	Last pixel in the scan line.
ST	Process	I	First pixel in the scan line.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The PBORDR subroutine is called by subroutines HDT and LANSAT (see section 3.4 and 3.5, respectively).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
NEXTP	Flag which, if = TRUE, instructs program to process next patch.

### 3.4.3 UNIT 4, LEVEL 2 - LDCODE

The LDCODE subroutine decodes image data from a foreign imagery tape in one of the following tape formats: Universal, Landsat 1 and 2, or LARSYS II and III.

CALLING SEQUENCE: SUBROUTINE LDCODE(LUN,IF,BUFFER,BFSZ,DLIN,RCHAN,LSS,LSE,BUF,EOF,PC,IE,INIT)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN1	In/out	I	Logical unit number assigned to the tape.
IF	Process	I	Format type of the tape.
BUFFER	Array	Byte	Array containing the scan line data and scratch pad buffer.
BFSZ	Process	I	Size of array buffer.
DLIN	Process	I	Requested line number; starts with line 1.
RCHAN	Process	I	Requested channel number, starts with 1.
LSS	Process	I	Pixel with which to start data move; assumes first pixel on tape.
LSE	Process	I	Last pixel desired.
BUF	Array	Byte	Decoded pixels LSS through LSE for calling program.
EOF	Process	I	Flag which, if = 1, indicates end of file; if = 0, indicates more data is to come.
PC	Process	I	Parity check counter.
IE	Process	I	Error code.
INIT	Process	I	Initialization flag which if = 0, indicates first pass; if = 1, not first pass.

#### FILES:

<u>Name</u>	<u>Usage</u>	<u>Record Format</u>
Tape	Input	Data tape being read.

COMMON BLOCKS: Common blocks for the LDCODE subroutine are /HCOM/ and /LABEL/. For a description of each, refer to sections 3.1 and 3, respectively.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The LDCODE subroutine is called by subroutines HDT and LANSAT (see sections 3.4 and 3.5, respectively).

LOCAL VARIABLES: Local variables for the LDCODE subroutine are listed in common block /HCOM/ in the subroutine HDREAD (section 3.1).

#### 3.4.4 UNIT 4, LEVEL 2 - PARTP

The PARTP subroutine adjusts the parameters when a patch is divided between two CCT strips.

CALLING SEQUENCE: Subroutine PARTP(PTCH,STARTP,STOPP,STOP,SEQNO,SE,ST,PIXELS,I,PATCH)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
PTCH	Flag	L	Flag which determines if the patch counter needs to be decremented.
STARTP	Process	I	Starting pixel number provided by the calling routine; altered if necessary for each patch.
STOPP	Process	I	Stop pixel for each patch; altered portion of patch is on the next CCT.
STOP	Process	I	Stop pixel for each patch.
SEQNO	Process	I	Sequence number for each CCT.
SE	Process	I	Last pixel in the scan line for the current CCT.
ST	Process	I	First pixel in the scan line for the current CCT.
PIXELS	Array	I	Array containing the desired patch pixels.
I	Process	I	Band number 1 through 4.
PATCH	Process	I	Patch number for resetting counter if necessary.

FILES: Not applicable.

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The PARTP subroutine is called by subroutines HDT and LANSAT (see section 3.4 and 3.5, respectively).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
CCTNO	The CCT strip number that is last stored.
PIECE	Array containing the parameter values to be used for the remainder of the rightmost patch located on the next CCT strip.

### 3.4.5 UNIT 4, LEVEL 2 - HIST

The HIST subroutine builds histograms for each channel.

CALLING SEQUENCE: Subroutine HIST(BUF,STARTP,STOPP,RCHAN,ST)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
BUF	Array	Byte	Array containing the scan line from the full frame.
STARTP	Process	I	Starting pixel number provided by the calling routine.
STOPP	Process	I	Stop pixel for each patch.
RCHAN	Process	I	Current channel number.
ST	Process	I	Starting pixel number for the current CCT.

FILES: Not applicable.

COMMON BLOCKS: Common blocks for the HIST subroutine are /HIST/ and /LABEL/. For a discription of each, refer to routine READTP (section 3).

SUBROUTINES CALLED: Not applicable.

CALLED BY: The HIST subroutine is called by subroutines HDI and LANSAT (see sections 3.4 and 3.5).

LOCAL VARIABLES: Not applicable.

### 3.5 UNIT 5 - LANSAT

The LANSAT subroutine extracts patch images from the full-frame image, with processing taking place one strip of CCT at a time. It further builds a 16-by-16 patch frame image consisting of 32-sample by 32-line pixel patches. Each pixel in a patch is comprised of bands 4 through 7. The subroutine uses Landsat 1 and 2 full-frame size images.

CALLING SEQUENCE: SUBROUTINE LANSAT(LUN1,LUN2,BUF,BFSZ,FLNIMD,EOF,IE,BUFFER,IFRM,SS,INIT)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN1	I/O	I	Logical unit number assigned to the tape.
LUN2	I/O	I	Logical unit number assigned to the disk file.
BUF	Array	I	Buffer containing the pixel data per scan line.
BFSZ	Process	I	Buffer size to be used.
FLNIMD	Array	I	Array containing the file name of the output file.
EOF	Process	I	End of file.
IE	Process	I	Error code.
BUFFER	Array	I	Scratch pad array containing header or image data.
IFRM	Process	I	Format type of the tape.
SS	Process	I	Start pixel number.
INIT	Process	I	Initializing flag.

For further information on the LANSAT subroutine, consult the documentation written for subroutine HDT (see section 3.4).

### 3.6 UNIT 6 - HDRHIS

The HDRHIS subroutine writes the histograms for each band (4-7) to the IMDACS header record file.

CALLING SEQUENCE: SUBROUTINE HDRHIS(LUN2,FLNIMH,UIFHR)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN2	I/O	I	Logical unit number assigned to a file on the disk.
FLNIMH	Array	A	Array containing the file name.
UIFHR	Array	A,I	Array containing the header data.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record Format</u>
??????.IMH	Header file	Consult IMDACS documentation for information on format.

#### COMMON BLOCKS:

<u>Name</u>	<u>Relative word number</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
H4	512	Array	R	Histogram for band 4.
H5	1025	Array	R	Histogram for band 5.
H6	1537	Array	R	Histogram for band 6.
H7	2049	Array	R	Histogram for band 7.

SUBROUTINE CALLED: Not applicable.

CALLED BY: The HDRHIS subroutine is called by the driver routine READTP (see section 3).

LOCAL VARIABLES: Not applicable.



#### 4. SUBSYSTEM 2 - SKIP

The SKIP routine is the driver routine for the skipped image generation program.

CALLING SEQUENCE: Not applicable.

CALLING ARGUMENTS: Not applicable.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record format</u>
Input	Cards	Card 1, columns 1-6; File name. Card 2, columns 1-2; Line start number. Card 3, columns 1-2; Pixel start number.

COMMON BLOCKS: Common blocks for the SKIP routine are COMH, LABEL, and HIST. Refer to driver routine READTP (section 3) for a description of each.

#### SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
ASNLUN	Assign logical unit number.	System
CLOSS	Close.	System
GETADR	Get address.	System
HDREAD	Header record read.	3
HDRFLN	Header record file name.	3.3
HDRHIS	Header record histogram.	3.6
OPENS	Open	System.

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
QIO	Queue I/O.	System
SKPRD	Skip read.	2
<u>CALLED BY:</u> Not applicable.		

#### 4.1 UNIT 1, LEVEL 1 - SKPRD

The SKPRD subroutine controls the reading and writing of the data to be selected from the full Landsat frame for the skipped image.

CALLING SEQUENCE: CALL SKPRD(LUN1,LUN2,BUF,BFSZ,EOF,IE,BUFFER,IFRM,HDRADR,ICCT,LINST,IPIXST)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN1	I/O	I	Logical unit for input tape.
LUN2	I/1	I	Logical unit for output data files.
BUF	Array	Byte	Buffer for pixel data.
BFSZ	Process	I	Size of buffer BUF.
EOF	Process	I	End of file indicator for tape read; if = 1, end of file is found.
IE	Process	I	Error indicator for tape read; if = 0, no error.
BUFFER	Array	Byte	Buffer for reading data from tape to be used by LDCODE.
IFRM	Process	I	Format of input tape; IFRM = 2 for LANSAT.
HDRADR	Process	I	Address of buffer.
ICCT	Process	I	The CCT strip number that is being processed.
LINST	Process	I	Start line number for building image.
IPIXST	Process	I	Pixel start number for building image.

FILES: Not applicable.

COMMON BLOCKS: The common block for the SKIP routine is /HIST/. Refer to the driver routine READTP (section 3) for a description of this common block.

SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
LDCODE	Line decode	4.1.1
SKPWRT	Skip write	4.1.2

CALLED BY: The SKPRD subroutine is called by the driver routine SKIP (see section 4).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
ILINE	Line number to be read from tape.
LINE	Line number to write to disk file.
LSS	Pixel start for CCT strip; always equal 1.
LSE	Pixel stop for CCT strip; equals number of pixels per scan on this strip.
IPIXL	First pixel to remove from this strip.
IPADD	Skip factor to add for pixel skipping; skip factors are 6, 6, 6, and 7.
LADD	Skip factor to add for line skipping; and skip factors are 4 and 5.
ISTR	Index into array IPADD; assures that the skip factor is not changed across CCT strips.

#### 4.1.2 UNIT 1, LEVEL 2 - SKPWRT

The SKPWRT routine writes the selected pixels for the skipped image to the output disk file.

CALLING SEQUENCE: CALL SKPWRT(LUN2,BUF,INDX,ICHAN,ICCT,LINE)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN2	I/O	I	Logical unit number for output data file.
BUF	Array	Byte	Buffer containing pixel data to output.
INDX	Process	I	Count of number of entries to process from BUF.
ICHAN	Process	I	Channel number for data in BUF.
ICCT	Process	I	The CCT strip number to process.
LINE	Process	I	Line number to write data in BUF.

#### FILES:

<u>File</u>	<u>Usage</u>	<u>Record format</u>
FILENM.IMD	Image data	IMDACS data file format.

COMMON BLOCKS: Not applicable.

#### SUBROUTINES CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
SKPHST	Skip histogram	4.1.2.1

CALLED BY: The SKPWRT subroutine is called by routine SKPPD (see section 4.1).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
ICNT	Count of number of pixels written to scan line for previous CCT strips; if ICCT = 1, then ICNT = 0.
OBUF	Output buffer to write to the file.

#### 4.1.2.1 UNIT 1, LEVEL 3 - SKPHST

The SKPHST subroutine computes the histogram for the skip image program.

CALLING SEQUENCE: CALL SKPHST(OBUF)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
OBUF	Array	Byte	Buffer containing pixel data to write skipped image data file.

FILE: Not applicable.

COMMON BLOCKS: The common block for subroutine SKPHST is HIST. Refer to the driver routine READTP (section 3) for a description of this common block.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The SKPHST subroutine is called by subroutine SKPWRT (see section 4.1.2).

LOCAL VARIABLES: Not applicable.

## 5. SUBSYSTEM 3 - PATCHP

The PATCHP program is the driver for the patch processor and produces analyst aid statistics for the 32-sample by 32-line pixel patches in a 512-sample by 512-line pixel image file.

CALLING SEQUENCE: Not applicable.

CALLING ARGUMENTS: Not applicable.

### FILES:

<u>File</u>	<u>Usage</u>	<u>Record format</u>
INPUT	CARDS	Card 1, columns 1-30; File name for header record file = FILENM.IMH.  Card 2, columns 1-30; File name for image data file = FILENM.IMD.
FILENM.IMH	Input	IMDACS image header record file.
FILENM.IMD	Input	IMDACS image data file.
FILENM.DAT	Output	Analyst aid statistics file.

### HEADER RECORD

<u>Columns</u>	<u>Format</u>	<u>Contents</u>
1	1X	Blank.
2-11	10A1	Scene identification.
12-13	I2	Sun angle.
14	I1	Satellite number.
15	A1	Latitude direction north or south.
16-19	I4	Latitude degrees.
20-21	I2	Latitude minutes.
22	A1	Longitude direction east or west.



<u>Columns</u>	<u>Format</u>	<u>Contents</u>
23-26	I4	Longitude degrees.
27-28	I2	Longitude minutes.
29-30	2X	Blank.
31-36	A6	File name.

DATA RECORD - 256 RECORDS, ONE RECORD PER PATCH

<u>Columns</u>	<u>Format</u>	<u>Contents</u>
1	1X	Blank.
2-4	I3	Patch number.
5-28	4F6.2	Channel means.
29-32	I4	Garbled pixel count.
33-36	I4	Cloud pixel count.
37-40	I4	Water pixel count.
41-46	F6.3	GAMA.
47-49	I3	Soil line number.
50-53	F4.1	Green number.
54-57	F4.1	AVI.
58-61	F4.1	PVI.
62-66	F5.3	TVI7.
67-71	F5.3	LAI.
72-74	I3	GIN5.
75-77	I3	GIN15.
78-80	I3	GIN30.

COMMON BLOCK: The common block for program PATCHP is PATCH.

# PATCH

<u>Name</u>	<u>Relative word number</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
CH	1-4096	Array	I,A	Array dimensioned (1024,4) containing 1024 pixel values for each of 4 channels belonging to a 32-by-32 patch.
KH	4096-9192	Array	I,A	Array dimensioned (1024,4) containing 1024 Kauth-transformed pixel values for each of 4 channels belonging to a 32-by-32 patch.
FLAG	9193-9704	Array	Byte	Array dimensioned (1024) to contain a flag value for each pixel after screening algorithm has been performed on a patch image; value of 1 = good pixel, value of 10-29 = water pixel, value of 30-39 = cloud pixel, value of 40-49 = garbled pixel.
X	9705-9720	Array	Real*4	Four-member array containing mean pixel value for each good pixel on each of 4 channels per patch.
GAR	9721-9722	Process	I*A	Count of garbled pixels in patch.
CLOUD	9723-9724	Process	I*A	Count of cloud pixels in patch.
WATER	9725-9726	Process	I*A	Count of water pixels in patch.
GAM	9727-9730	Process	Real	Integer form of GAMA.
VEG	9731-9778	Array	Real	Array dimensioned (12) to save vegetative indices.
GIN	9779-9784	Array	I*A	Array dimensioned (3) to contain values for GIN5, GIN15, and GIN30.
IAG	9785-9786	Process	I*A	Sun angle.
SAT	9787-9788	Process	I*A	Satellite number.
SLN	9789-9790	Process	I*A	Soil line number.
GAMA	9791-9794	Process	Real	X-STAR haze parameter.

SUBROUTINES CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
CLOS\$	Close	System
GAMMA	X-STAR depth correction	5.3
HAZCOR		5.4
HEADER	Header process	5.1
OPENS	Open	System
PATCRD	Patch read	5.2
PATRPT	Patch report	5.6
VIN	Vegetation index	5.5

CALLED BY: Not applicable.

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
IPATC <sub>i</sub>	Patch number being processed (1-256).

### 5.1 UNIT 1, LEVEL 1 - HEADER

The HEADER subroutine reads the universal header record from the image header file. It decodes some information from it, and writes it for the header record of the output analyst aid statistics file.

CALLING SEQUENCE: CALL HEADER(IMHNAM,FILANAM,BUF,IAG,SAT,IE)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
IMHNAM	Array	Byte	Name of IMDACS image header file.
FILNAM	Array	Byte	Name of analyst aid statistics output file.
BUF	Array	Byte	Array to read universal header record into record 2 of header file.
IAG	Process	I*2	Sun angle.
SAT	Process	I*2	Satellite number.
IE	Process	I*2	Error indicator; if error is found, set to 1.

FILES: See program PATCHP (section 5) for files information.

COMMON BLOCKS: Not applicable.

#### SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Level</u>
CLOS\$	Close	System

CALLED BY: The HEADER subroutine is called by the program PATCHP (see section 5).

LOCAL VARIABLES: Not applicable.

## 5.2 UNIT 1, LEVEL 1 - PATCHRD

The PATCHRD subroutine reads four channels of imagery data for one patch into an integer \*4 buffer.

CALLING SEQUENCE: CALL PATCHRD(LUN,IPATCH,BUF)

### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LUN	I/O	I*2	Logical unit number for IMDACS image data file.
IPATCH	Process	I*2	Patch number for which data is desired.
BUF	Array	I*2	(1024,4) dimensioned array for imagery data.

FILES: The PATCHRD routine reads the IMDACS image data file (for description, see PATCHP, section 5).

COMMON BLOCKS: Not applicable.

SUBROUTINES CALLED: Not applicable.

CALLED BY: The PATCHRD subroutine is called by the PATCHP program (see section 5).

### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
ILIN	Line number within image file where requested patch data starts.
IPIX	Pixel number within ILIN where requested patch data starts.

### 5.3 UNIT 1, LEVEL 1 - GAMMA

The subroutine GAMMA computes the X-STAR optical depth parameter.

CALLING SEQUENCE: CALL GAMMA

CALLING ARGUMENTS: Not Applicable.

FILES: Not applicable.

COMMON BLOCKS: The common block for the GAMMA subroutine is PATCH (see section 5).

#### SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
CALIB	Calibrate mean	5.3.1
MEAN	Calculate mean	5.3.2
SCRAL	Screen	5.3.3
\$SIN	Sine function	System

CALLED BY: The GAMMA subroutine is called by the PATCHP program (see section 5).

#### LOCAL VARIABLES.:

<u>Name</u>	<u>Definition</u>
CR	Sun-angle correction factor; corrects to 52 degrees.
2	Sun-angle correction applied to each pixel value.

### 5.3.1 UNIT 1, LEVEL 2 - CALIB

The CALIB subroutine applies calibration constants to the pixel data. The calibration constants are dependent on the satellite number in the following way, where LK = the satellite correction constant from which to choose:

- LK = -1 for postlaunch Landsat 2
- LK = 0 for no correction
- LK = 1 for Landsat 1 calibration
- LK = 2 for prelaunch Landsat 2 (no correction)
- LK = 3 for LEC Landsat 3 calibration
- LK = 4 for ERIM Landsat 3 calibration

CALLING SEQUENCE: CALL CALIB(LK,N)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
LK	Process	I*2	Indicates satellite correction constant to choose.
N	Process	I*2	Number of possible pixel values to be calibrated equals 1024.

FILES: Not applicable.

COMMON BLOCKS: The common block for the CALIB subroutine is PATCH (see section 5).

SUBROUTINE CALLED: Not applicable.

CALLED BY: The CALIB subroutine is called by the subroutine GAMMA (see section 5.3).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>			
CALFAC	Calibration factor array (channel, LK), as follows:			
	1.040	1.000	1.090	0.820
	1.275	1.141	1.098	0.948
	1.161	1.230	1.246	1.062
	1.137	1.173	1.247	1.126
CALBAS	Calibration bias array (channel, LK), as follows:			
	-5.790	1.190	-2.910	3.010
	-1.445	-2.712	-2.950	0.446
	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0



### 5.3.2 UNIT 1, LEVEL 2 - MEAN

The MEAN subroutine computes the mean and standard deviation for a requested channel in the patch image. Only good pixels will be used in the computations.

CALLING SEQUENCE: CALL MEAN(ICHAN,N,NN,STDEV)

CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
ICHAN	Process	I*2	Channel for which calculations are requested.
N	Process	I*2	Number of possible pixel values to use in computation; equals 1024.
NN	Process	I*2	Count of number of pixel values actually used in computations.
STDEV	Process	I*2	Standard deviation for channel requested.

FILES: Not applicable.

COMMON BLOCKS: The common block for subroutine MEAN is PATCH (see section 5).

SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
\$SORT	Square root	System

CALLED BY: The MEAN subroutine is called by GAMMA subroutine GAMMA (see section 5.3).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
S	Sum of good pixel values.
SS	Sum of square of good pixel values.
VARI	Variance on requested channel.

### 5.3.3 UNIT 1, LEVEL 2 - SCRALL

The subroutine SCRALL uses the Kauth transformation and screens the pixel data and calculates GIN5, GIN15, and GIN30.

CALLING SEQUENCE: CALL SCRALL(N)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
N	Process	I*2	Number of pixels to screen equals 1024.

FILES: Not applicable.

COMMON BLOCKS: The common block for subroutine SCRALL is PATCH (see section 5).

#### SUBROUTINE CALLED:

<u>Name</u>	<u>Definition</u>	<u>Level</u>
\$SIN	Sine computation	System

CALLED BY: Subroutine SCRALL is called by GAMMA (see section 5.3 for description).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
RR	Kauth transformation.
2	Screening algorithm test vectors.
CC	Screening algorithm thresholds.
II, JJ	Screening algorithm pointers.
LL	Screening algorithm output values.

#### 5.4 UNIT 1, LEVEL 1 - HAZCOR

The HAZCOR subroutine applies the XSTAR haze correction that is given to GAMMA and the Sun angle.

CALLING SEQUENCE: CALL HAZCOR

CALLING ARGUMENT: Not applicable.

FILES: Not applicable.

COMMON BLOCKS: The common block for the HAZCOR subroutine is PATCH (see section 5 for a description).

#### SUBROUTINES CALLED:

<u>Name</u>	<u>Definition</u>	<u>Reference (section)</u>
KAUTH	Kauth transform	5.4.1
\$EXP	Exponent	System
\$SIN	Sine function	System

CALLED BY: The HAZCOR subroutine is called by the driver routine PATCHP (see section 5).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
COR	Sun-angle correction factor.
A	Gain for correction.
B	Bias for correction.

#### 5.4.1 UNIT 1, LEVEL 2 - KAUTH

The KAUTH subroutine computes the Kauth Thomas data transformation for the patch image.

CALLING SEQUENCE: CALL KAUTH(N)

#### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
N	Process	I*2	Number of pixels to transform equals 1024.

FILES: Not applicable.

COMMON BLOCKS: The common block for the KAUTH subroutine is PATCH (see section 5 for description).

SUBROUTINES CALLED: Not applicable.

CALLED BY: The KAUTH subroutine is called by routine HAZCOR (see section 5.4).

#### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
	Kauth matrix, as follows:
R	.332    .603    .676    .263
	-.283   -.660    .577    .388
	-.900    .428    .076   -.040
	-.016    .131   -.452    .882

## 5.5 UNIT 1, LEVEL 1 - VIN

The subroutine VIN calculates a requested vegetative index number. The following list describes the vegetative indexes that can be requested and their request identification (ID) number.

<u>ID number</u>	<u>Vegetative index</u>	<u>Computation</u>
1	Greenness	$-.283*CH1-.660*CH2+.577*CH3+.388*CH4$
2	Green number	$(-.283*CH1-.660*CH2+.577*CH3+.388*CH4)+5$
3	<sup>a</sup> AVI	$2*CH4-CH2$
4	<sup>b</sup> PVI	$ABS(.38425*(2.4*CH4-CH2))$
5	<sup>c</sup> TVI6	$SQRT(ABS((CH3-CH2)/(CH3+CH2)+.5))$
6	<sup>d</sup> TVI7	$SQRT(ABS((2*CH4-CH2)/(2*CH4+CH2)+.5))$
7	Leaf Area Index - Simple Form	$1.653*CH1/CH2-1.698*CH1/CH3+.093$
8	Leaf Area Index - Complex form	$2.677-3.694*CH1/CH2-2.309*CH1/CH3$ $+2.875*CH1/CH4+.043*CH2/CH3$ $-1.346*CH2/CH4+3.017*((CH1/CH2)$ $-(CH1/(2*CH4)))*CH1/CH2$
9	Berkeley	$2*CH4/CH2$
10	<sup>e</sup> DVI	$CH2-2.4*CH4$
11	<sup>f</sup> RVI	$CH2/CH4$
12	Unused	

<sup>a</sup> Ashburn Vegetative Index

<sup>b</sup> Perpendicular Vegetative Index

<sup>c</sup> Transformed Vegetative Index 6

<sup>d</sup> Transformed Vegetative Index 7

<sup>e</sup> Difference Vegetative Index

<sup>f</sup> Red Vegetative Index

CALLING SEQUENCE: CALL VIN(ID,N)

CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
ID	Process	I*2	One of the 12 identification numbers listed above.
N	Process	I*2	Number of possible pixel values to use in computation equals 1024.

FILES: Not applicable.

COMMON BLOCKS: The common block for the VIN subroutine is PATCH (see section 5 for a description).

SUBROUTINES CALLED:

<u>Name</u>	<u>Definition</u>	<u>Level</u>
\$SQRT	Square root	System

CALLED BY: The VIN subroutine is called by PATCHP (see section 5 for a description).

LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
ICNT	Count of pixels used in calculating requested vegetative index.

## 5.6 UNIT 1, LEVEL 1 - PATRPT

The PATRPT subroutine formats the analyst aid statistics file and writes a printer report of the contents.

CALLING SEQUENCE: CALL PATRPT(FILNAM)

### CALLING ARGUMENTS:

<u>Name</u>	<u>Usage</u>	<u>Type</u>	<u>Definition</u>
FILNAM	Array	Byte	File name of the analyst aid statistics file to be output as a report.

FILES: This PATRPT subroutine reads the analyst aid statistics file documented in section 5.

COMMON BLOCK: Not applicable.

### SUBROUTINES CALLED:

<u>Name</u>	<u>Meaning</u>	<u>Level</u>
CLOS\$	Close	System
OPEN\$	Open	System

CALLED BY: The PATRPT subroutine is called by the driver routine PATCHP (see section 5).

LOCAL VARIABLES: Not applicable.



## 6. POSPP

The post patch processor program POSTPP computes the spatial response function by regression onto a quadratic surface.

CALLING SEQUENCE: Not applicable.

CALLING ARGUMENTS: Not applicable.

### FILES:

<u>File</u>	<u>Usage</u>	<u>Record format</u>
FILENM.DAT	Analyst aid Statistics file;	see PATCHP (section 5) for format.

COMMON BLOCKS: Not applicable.

### SUBROUTINES CALLED:

<u>Name</u>	<u>Meaning</u>	<u>Level</u>
OPEN\$	Open	System

CALLED BY: Not applicable.

### LOCAL VARIABLES:

<u>Name</u>	<u>Definition</u>
ISUM	Inaccurate pixel count.
RN**	Normalizing constants.
P(K)	Orthogonal polynomials.
C(K)	Raw orthogonal coefficients.
VINH	Estimated variable value.
R(I,J)	Residual computation.
SSR	Sum of square residual.

<u>Name</u>	<u>Definition</u>
SUM	Sum of residuals.
RMSE	Root mean square error.
CN**	Coefficients for orthogonal polynomials.
SAS**	Coefficients for simple polynomials.
VAR	Variable of interest.

## 7. REFERENCES

1. Lambeck, P. F.: Implementation of the X-star Haze Correction Algorithm and Associated Preprocessing Stats for Landsat Data. ERIM IS-PFL-1272.

## APPENDIX A

### LISTING OF PATCH PROCESSOR

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:35:06	30-OCT-80	PAGE 1
PATCHP. FTN		/TR: BLOCKS/WR		
0001		PROGRAM PATCHP		
	C			
	C	THIS PROGRAM WILL BE THE DRIVER FOR THE PATCH PROCESSOR		
	C			
0002		COMMON/PATCH/CH,KH,FLAG,X,GAR,CLOUD,WATER,GAM,VEG,GIN,IAG, 1 SAT,SLN,GAMA		
0003		INTEGER*2 IAG,SAT,CH(1024,4),KH(1024,4)		
0004		INTEGER*2 GAR,CLOUD,WATER,GIN(3),SLN		
0005		BYTE FLAG(1024)		
0006		REAL GAM,X(4),VEG(12),GAMA		
0007		BYTE FILNAM(30),IMHNM(30),IMDNAM(30)		
	C			
	C	READ NAME OF HEADER FILE AND IMAGE DATA FILE		
	C			
0008		READ(5,1000)IMHNM		
0009	1000	FORMAT(30A1)		
0010		DO 10 I=1,30		
0011		IF(IMHNM(I).NE.' ')GO TO 10		
0012		IMHNM(I)=0		
0013		GO TO 15		
0014	10	CONTINUE		
0015	15	READ(5,1000)IMDNAM		
0016		DO 20 I=1,30		
0017		IF(IMDNAM(I).NE.' ')GO TO 20		
0018		IMDNAM(I)=0		
0019		GO TO 25		
0020	20	CONTINUE		
	C			
	C	CONSTRUCT NAME OF OUTPUT FILE USE NAME OF IMD FILE BUT		
	C	SUBSTITUTE DAT FOR IMD		
	C			
0021	25	DO 30 I=1,30		
0022		FILNAM(I)=IMDNAM(I)		
0023		IF(IMDNAM(I).EQ.' ')GO TO 40		
0024	30	CONTINUE		
0025	40	FILNAM(I+1)='D'		
0026		FILNAM(I+2)='A'		
0027		FILNAM(I+3)='T'		
0028		FILNAM(I+4)=0		
0029		IE=0		
0030		CALL HEADER(IMHNM,FILNAM,CH(1,1),IAG,SAT,IE)		
0031		IF(IE.NE.0)GO TO 990		
	C			
	C	OPEN IMAGE DATA FILE		
	C			
0032		OPEN(UNIT=1,NAME=IMDNAM,TYPE='OLD',SHARED,READONLY, 1 ACCESS='DIRECT',ERR=900)		
	C			
12	C	LOOP FOR PROCESSING PATCHES		
11	C			
10	0033	DO 100 IPATCH=1,256		
9	C	WRITE(5,1235)IPATCH		
8	0034	FORMAT(1X,'PROCESSING PATCH ',15)		
7	0035	CALL PATCHRD(1,IPATCH,CH(1,1))		
6	0036	CALL GAMMA		
5	0037	CALL HAZCOR		
4				
3				
2				
1				

FORTRAN IV-PLUS V02-51E		14:35:06	30-OCT-80	PAGE 2
PATCHP.FTN		/TR:BLOCKS/WR		
0038		CALL VIN(2,1024)		
0039		CALL VIN(3,1024)		
0040		CALL VIN(4,1024)		
0041		CALL VIN(6,1024)		
0042		CALL VIN(7,1024)		
0043		WRITE(2,1100)IPATCH,(X(I),I=1,4),GAR,CLOUD,WATER,GAMA,SLN,VEG(2),		
		1 VEG(3),VEG(4),VEG(5),VEG(7),(GIN(I),I=1,3)		
0044	1100	FORMAT(1X,13,4F6.2,3I4,F6.3,13,3F4.1,2F5.3,3I3)		
0045	100	CONTINUE		
	C			
	C	FINISHED		
	C			
0046		CLOSE(UNIT=1)		
0047		CLOSE(UNIT=2)		
	C			
	C	WRITE PATCH REPORT		
	C			
0048		CALL PATRPT(FILNAM)		
0049		GO TO 990		
	C			
	C	ERROR		
	C			
0050	900	WRITE(6,1200)		
0051	1200	FORMAT(1X,'ERROR IN OPENING IMAGE DATA FILE')		
0052	990	CONTINUE		
0053		END		

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1 001130 300		RM.I, CON, LCL
2	\$PDATA 000034 14		RM.D, CON, LCL
3	\$IDATA 000270 92		RM.D, CON, LCL
4	\$VARS 000140 48		RM.D, CON, LCL
6	PATCH 042132 8749		RM.D, OVR, GBL

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CLOUD	I*2	6-042032	GAM	R*4	6-042024	GAMA	R*4	6-042126
LAG	I*2	6-042120	IE	I*2	4-000134	IPATCH	I*2	4-000136
WATER	I*2	6-042024				SAT	I*2	6-042122
						SLN	I*2	6-042124

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
CH	I*2	6-000000	020000	4096 (1024, 4)
FILNAM	L*1	4-000000	000036	15 (30)
FLAG	L*1	6-040000	002000	512 (1024)
GIN	I*2	6-042112	000006	3 (3)
IMDRIAM	L*1	4-000074	000036	15 (30)
IMHNIAM	L*1	4-000036	000036	15 (30)
KH	I*2	6-020000	020000	4096 (1024, 4)
VEG	R*4	6-042032	000060	24 (12)
X	R*4	6-042000	000020	8 (4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000114	15	1-000136	20	1-000232	25	1-000254
40	1-000340	100	**	900	1-001070	990	1-001116
1100	3-000004	1200	3-000040	1235	**	1000	3-000000

FUNCTIONS AND SUBROUTINES REFERENCED

CLOS\$ GAMMA HAZCOR HEADER OPENS\$ PATCRD PATRPT VIN

11 TOTAL SPACE ALLOCATED = 043746 9203

10 NO FPP INSTRUCTIONS GENERATED

7 .TY17 (320.40)A, B/-SP=SY0: (320.40)PATCHP

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:05:22	30-OCT-80	PAGE 1
HEADER FTN		/TR:BLOCKS/WR		
0001	SUBROUTINE HEADER(IMHNAM, FILNAM, BUF, IAG, SAT, IE)			
	C	THIS SUBROUTINE WILL READ THE UNIVERSAL HEADER RECORD		
	C	FROM THE IMH FILE. IT WILL DECODE SOME INFO AND WRITE IT TO		
	C	AN OUTPUT .DAT FILE		
	C	CALL ARGUMENTS:		
	C	IMHNAM	I	NAME OF IMH FILE
	C	FILNAM	I	NAME OF OUTPUT DATA FILE
	C	BUF	I	BUFFER TO USE TO READ HEADER DATA INTO
	C	IAG	O	SUN ANGLE OF ELEVATION
	C	SAT	O	SATELLITE NUMBER
	C	IE	O	ERROR INDICATOR SET TO 1 FOR ERROR
	C			
0002	BYTE IMHNAM(30), FILNAM(30), BUF(3072), BWORD(2)			
0003	INTEGER SAT			
0004	EQUIVALENCE (IWORD, BWORD)			
0005	OPEN(UNIT=1, NAME=IMHNAM, ACCESS='DIRECT', READONLY,			
		1 SHARED, TYPE='OLD', ERR=100)		
	C	READ HEADER AND DECODE DATA		
	C			
0006	IWORD=0			
0007	READ(1:2)(BUF(I), I=1, 3060)			
	C	READ SUN ANGLE OF ELEVATION BYTES 1833-1834		
	C			
0008	BWORD(1)=BUF(1834)			
0009	BWORD(2)=BUF(1833)			
0010	IAG=IWORD			
	C	READ SATELLITE NUMBER BYTES 65-66		
	C			
0011	BWORD(1)=BUF(66)			
0012	BWORD(2)=BUF(65)			
0013	SAT=IWORD			
	C	READ LATITUDE AND LONGITUDE BYTES 1985-1995		
	C			
0014	BWORD(1)=BUF(1987)			
0015	BWORD(2)=BUF(1986)			
0016	LATD=IWORD			
0017	BWORD(1)=BUF(1989)			
0018	BWORD(2)=BUF(1988)			
0019	LATM=IWORD			
0020	BWORD(1)=BUF(1992)			
0021	BWORD(2)=BUF(1991)			
0022	LOND=IWORD			
12 0023	BWORD(1)=BUF(1994)			
11 0024	BWORD(2)=BUF(1993)			
10 0025	LONM=IWORD			
9 0026	CLOSE (UNIT=1)			
8	C			
7	C	WRITE DATA TO OUTPUT FILE		
6	C			
5 0027	OPEN (UNIT=2, NAME=FILNAM, TYPE='NEW', ERR=200,			
4				
3				



```

FORTRAN IV-PLUS V02-51E      14 35:22      30-OCT-80      PAGE 2
HEADER.FTN      /TR:BLOCKS/WR

      1 FORM='FORMATTED')
0028      WRITE(2,1000)(BUF(I),I=33,42),IAG,SAT,BUF(1985),LATD,LATM,
      1 BUF(1990),LOND,LONM,(FILNAM(I),I=1,6)
0029      1000      FORMAT(1X,10A1,I2,I1,A1,I4,I2,A1,I4,I2,2X,6A1,44X)
      C
      C      CHECK FOR GOOD DATA FROM HEADER
      C
0030      IF(IAG.LT.15.OR.IAG.GT.75)IE=1
0031      IF(SAT.LT.1.OR.SAT.GT.3)IE=1
0032      IF(IE.EQ.1)WRITE(6,1300)IAG,SAT
0033      1300      FORMAT(1X,'ERROR IN SUN ANGLE =',I5,2X,'OR IN SATELLITE NUMBER =',
      1 I5)
0034      IF(SAT.EQ.2)SAT=-1
0035      RETURN
      C
      C      ERROR MESSAGES
      C
0036      100      WRITE(6,1100)
0037      1100      FORMAT(1X,'ERROR IN OPENING HEADER FILE')
0038      IE=1
0039      RETURN
0040      200      WRITE(6,1200)
0041      1200      FORMAT(1X,'ERROR IN OPENING OUTPUT DATA FILE')
0042      IE=1
0043      RETURN
0044      END

```

FORTRAN IV-PLUS V02-51E																14:35:22																30-OCT-80																PAGE 3															
HEADER FIN																71R BLOCKS/WR																																															
PROGRAM SECTIONS																																																															
NUMBER NAME SIZE																ATTRIBUTES																																															
1 *CODE1 001112 293																RW, I, CON, LCL																																															
3 *IDATA 000362 121																RW, D, CON, LCL																																															
4 *VARS 000014 6																RW, D, CON, LCL																																															
5 *TEMPS 000002 1																RW, D, CON, LCL																																															
ENTRY POINTS																																																															
NAME TYPE ADDRESS NAME TYPE ADDRESS NAME TYPE ADDRESS NAME TYPE ADDRESS																																																															
HEADER 1-000000																																																															
VARIABLES																																																															
NAME TYPE ADDRESS NAME TYPE ADDRESS NAME TYPE ADDRESS NAME TYPE ADDRESS																																																															
I I*2 4-000002 IAG I*2 F-000010* IE I*2 F-000012* IWRD I*2 4-000000 LATD I*2 4-000006 LOND I*2 4-000010 LGNM I*2 4-000012 SAT I*2 F-000012*																																																															
ARRAYS																																																															
NAME TYPE ADDRESS SIZE																DIMENSIONS																																															
BUF L*1 F-000006* 005000 1536																(3072)																																															
EJCFD L*1 4-000000 000002 1																(2)																																															
FILNAM L*1 F-000004* 000036 15																(30)																																															
INPRAM L*1 F-000002* 000036 15																(30)																																															
LABELS																																																															
LABEL ADDRESS LABEL ADDRESS LABEL ADDRESS LABEL ADDRESS LABEL ADDRESS																																																															
100 1-001016 200 1-001054 1000' 3-000000 1100' 3-000124 1200' 3-000164																																																															
1300' 2-000034																																																															
FUNCTIONS AND SUBROUTINES REFERENCED																																																															
CLOSING OPENING																																																															
TOTAL SPACE ALLOCATED = 001512 421																																																															
NO FPP INSTRUCTIONS GENERATED																																																															
.TT17. (320, 403A B/-SP-SY0: (320, 403)HEADER																																																															

ORIGINAL PAGE 12  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:35:38	30-OCT-80	PAGE 1
HAZCOR.FTN		/TR:BLOCKS/WR		
C HAZCOR.FTN				
C				
0001		SUBROUTINE HAZCOR		
0002		IMPLICIT INTEGER (A-Z)		
0003		COMMON/PATCH/CH,KH,FLAG,X,GAR,CLOUD,WATER,GAM,VEG,GIN,IAG, 1 SAT,SLN,GAMA		
0004		INTEGER*2 IAG,SAT,CH(1024,4),KH(1024,4)		
0005		INTEGER*2 GAR,CLOUD,WATER,GIN(3),SLN		
0006		BYTE FLAG(1024)		
0007		REAL GAM,X(4),VEG(12),GAMA		
0008		REAL A(4),B(4),COR,ANGRAD		
C				
D				
0009	100	TYPE 100		
		FORMAT(' HAZCOR DEBUG ON')		
0010		GAMA=(GAM-300)/1000		
0011		IF(GAMA.GT.0.7 .OR. GAMA.LT.-0.7) GO TO 333		
0012		IF(IAG.LT.15 .OR. IAG.GT.75) GO TO 333		
0013	1	ANGRAD=IAG*3.1416/180		
0014		COR=.77713/SIN(ANGRAD)		
0015		A(1)=EXP(-1.2680*GAMA)		
0016		A(2)=EXP(-1.0910*GAMA)		
0017		A(3)=EXP(-0.8358*GAMA)		
0018		A(4)=EXP(-0.5981*GAMA)		
0019		B(1)=61.9*(1-A(1))*COR		
0020		B(2)=66.2*(1-A(2))*COR		
0021		B(3)=83.2*(1-A(3))*COR		
0022		B(4)=33.9*(1-A(4))*COR		
0023		A(1)=A(1)*COR		
0024		A(2)=A(2)*COR		
0025		A(3)=A(3)*COR		
0026		A(4)=A(4)*COR		
D				
0027	102	TYPE 102,A,B		
		FORMAT(' HAZCOR A AND B',/ 8F6.3)		
C				
C				
0028		DO 2 K=1,1024		
0029		CH(K,1)=CH(K,1)*A(1)+B(1)		
0030		CH(K,2)=CH(K,2)*A(2)+B(2)		
0031		CH(K,3)=CH(K,3)*A(3)+B(3)		
0032		CH(K,4)=CH(K,4)*A(4)+B(4)		
0033	2	CONTINUE		
0034		CALL KAUTH(1024)		
C				
C				
C				
		SAT=-2		
		IAG=51		
		GAM=300		
0035		RETURN		
	C333	TYPE 200, GAM,GAMA,IAG,SAT		
0036	200	FORMAT(' HAZCOR PROBLEMS',/,		
12	*	GAM=', I6,/,		
11	*	GAMA=', F8.4,/,		
10	*	IAG=', I6,/,		
9	*	SAT=', I6,/,		
8	*	CR TO CONTINUE S TO STOP')		
7	C	ACCEPT 201,IIT		
6	0037	201	FORMAT(A1)	
5	0038	333	WRITE(6,2100)	
4				
3				

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14 35 38	30-OCT-80	PAGE 2
HAZCOR FTN		/TR. BLOCKS/WR		

0039	2100	FORMAT(1X, 'HAZCOR PROBLEMS ENCOUNTERED')
0040		IF(IIT.EQ.'S') STOP
0041		GO TO 1
0042		END

12  
11  
10  
9  
8  
7  
6  
5  
4  
3

ORIGINAL PAGE NO  
OF POOR QUALITY

FORTPAN IV-PLUS V02-51E 14:35:38 30-OCT-60 PAGE 3  
HAZCOR FTR /TR BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000574 222	RM, I, CON, LCL
2	\$DATA	000054 26	RM, D, CON, LCL
3	\$IDATA	000044 18	RM, D, CON, LCL
4	\$VARS	000054 22	RM, D, CON, LCL
6	PATCH	042132 8749	RM, D, DVR, GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
HAZCOR		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ARMSTRAN R*4		4-000044	CLOUD	I*2	6-042022	COR	R*4	4-000040
GAR	I*2	6-042020	IAG	I*2	6-042120	IT	I*2	4-000052
SLN	I*2	6-042124	WATER	I*2	6-042024			

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
A	R*4	4-000000	000030	8 (4)
B	R*4	4-000020	000030	8 (4)
CH	I*2	6-000000	020000	4096 (1024,4)
FLAG	L*1	6-040000	000000	512 (1024)
GIN	I*2	6-042112	003006	3 (3)
K*1	I*2	6-020000	020000	4096 (1024,4)
VEG	R*4	6-042032	000040	24 (12)
X	R*4	6-042000	000020	8 (4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000120	2	**	100'	**
201'	**	333	1-000624	2100'	3-000000

FUNCTIONS AND SUBROUTINES REFERENCED

5' FAUTH \$EXP \$SIN

TOTAL SPACE ALLOCATED = 043232 9037

ORIGINAL PAGE 11  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14 35 38 30-OCT-80 PAGE 4  
HAZCOR FTN /TR BLOCKS/WR

, TT17 [320, 40]A B/-SP=SYO [320, 40]HAZCOR

12  
11  
10  
9  
8  
7  
6  
5  
4  
3

3

ORIGINAL  
OF FOUR

```

FORTRAN IV-PLUS V02-51E      14:40:42   30-OCT-80      PAGE 1
PATRPT.FTN      /TR:BLOCKS/WR

C
C      SUBROUTINE PATRPT
C      PATCH REPORT WRITER
C
C      THIS PROGRAM WILL FORMAT THE FILE CREATED BY THE PATCH
C      PROCESSOR INTO A REPORT FORM.
C
C      SUBROUTINE PATRPT(FILNAM)
C      BYTE FILNAM(30), ID(10), NAME(6)
C      INTEGER GAR, CLOUD, WATER, GIN5, GIN15, GIN30, SLN
C
C      OPEN(UNIT=1, NAME=FILNAM, TYPE='OLD', FORM='FORMATTED')
C      READ(1, 1100) ID, IAG, ISAT, LATDR, LATD, LATM, LONDR, LONM, NAME
C      FORMAT(1X, 10A1, 12, I1, A1, 14, I2, A1, 14, I2, 2X, 6A1)
C
C      WRITE HEADER FOR REPORT
C
C      WRITE(6, 1200) ID, NAME
C      FORMAT(40X, 'PATCH REPORT FOR SCENE', 1X, 10A1, 2X, 'FILE: ', 6A1, 77)
C      WRITE(6, 1300) IAG, ISAT, LATDR, LATD, LATM, LONDR, LONM
C      FORMAT(5X, 'SUN ANGLE : ', 15, 5X, 'SATELLITE NUMBER : ', 15,
C      1 5X, 'LATITUDE : ', 1X, A1, 1X, 14, 'DEGREES', 1X, 12, 'MINUTES',
C      1 5X, 'LONGITUDE : ', 1X, A1, 1X, 14, 'DEGREES', 1X, 12, 'MINUTES', 77)
C      WRITE(6, 1400)
C      FORMAT(1X, 'PATCH', 8X, 'CHANNEL MEANS', 9X, 'GARBL', 2X, 'CLOUD',
C      1 1X, 'WATER', 3X, 'GAMA', 3X, 'SLN', 2X, 'GREEN', 2X, 'AVI', 3X, 'PVI', 4X,
C      2 'TVI7', 3X, 'LAI', 3X, 'GIN5', 1X, 'GIN15', 1X, 'GIN30', 77)
C
C      DO LOOP TO WRITE REPORT DATA
C
C
C      DO 100 I=1, 256
C      READ(1, 1500) IPATCH, X1, X2, X3, X4, GAR, CLOUD, WATER, GAMA, SLN, VEG2,
C      1 VEG3, VEG4, VEG6, VEG7, GIN5, GIN15, GIN30
C      FORMAT(1X, 13, 4F6, 2, 3I4, F6, 3, I3, 3F4, 1, 2F5, 5, 3I3)
C      WRITE(6, 1600) IPATCH, X1, X2, X3, X4, GAR, CLOUD, WATER, GAMA, SLN, VEG2,
C      1 VEG3, VEG4, VEG6, VEG7, GIN5, GIN15, GIN30
C      FORMAT(2X, 13, 2X, 4(F6, 2, 1X), 2X, 14, 3X, 14, 2X, 14, 1X, F6, 3, 2X, 13, 2X, F5, 1,
C      1 1X, F5, 1, 1X, F5, 1, 2X, F5, 3, 2X, F5, 3, 3X, 13, 2X, 13, 3X, 13)
C      CONTINUE
C      CLOSE (UNIT=1)
C      RETURN
C      END

```

14:40:42 30-OCT-60 PAGE 2

FORTRAN IV-PLUS V02-51E  
PATRPT.FTN /TR.BLOCKS/MR

PROGRAM SECTIONS

ATTRIBUTES

SIZE

1 \$CODE1 001172 317 RM.I.CON.LCL  
3 \$DATA 000736 239 RM.D.CON.LCL  
4 \$VARS 000132 45 RM.D.CON.LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
------	------	---------	------	------	---------	------	------	---------	------	------	---------

PATRPT 1-000003

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CLOUD	I*2	4-000032	GAMA	R*4	4-000102	OAR	I*2	4-000020	GIN30	I*2	4-000032
GIN5	I*2	4-000026	I	I*2	4-000056	IAG	I*2	4-000036	IPATCH	I*2	4-000060
LATD	I*2	4-000044	LATDR	I*2	4-000042	LATM	I*2	4-000046	LDND	I*2	4-000052
LONM	I*2	4-000054	SLN	I*2	4-000034	VEG2	R*4	4-000106	VEG3	R*4	4-000112
VEG6	R*4	4-000122	VEG7	R*4	4-000126	WATER	I*2	4-000024	X1	R*4	4-000062
X3	R*4	4-000072	X4	R*4	4-000076				X2	R*4	4-000066

ARRAYS

DIMENSIONS

FILNAM L\*1 F-000002\* 000036 15 (30)  
ID L\*1 4-000000 000012 5 (10)  
NAME L\*1 4-000012 000006 3 (6)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
15	**	160	**	1100'	3-000000	1200'	3-000032
1400'	3-000310	1500'	3-000312	1600'	3-000546	1300'	3-000110

FUNCTIONS AND SUBROUTINES REFERENCED

CLOS\* OPEN\*

TOTAL SPACE ALLOCATED = 002262 601

NO FPP INSTRUCTIONS GENERATED

TT17: [320.40]A B/-SP=SYO: [320.40]PATRPT



FORTRAN IV-PLUS V02-51E 14:35.54 30-OCT-80 PAGE 2  
PATCRD FTR 7TR:GLUCKS/MR

PROGRAM SECTIONS

NUMBER NAME SIZE ATTRIBUTES

1 \$CODE1 000360 120 RM.I,CON,LCL  
3 \$IDATA 000022 9 RM.D,CON,LCL  
4 \$VARS 004124 1066 RM.D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
------	------	---------	------	------	---------	------	------	---------	------	------	---------

PATCRD		1-000000									
--------	--	----------	--	--	--	--	--	--	--	--	--

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
I	I*2	4-004120	ILIN	I*2	4-004110	INDX1	I*2	4-004114	IPATCH	I*2	F-000004*
I*2	I*2	4-004112	I*2	I*2	4-000000	J	I*2	4-004112	LUN	I*2	F-000002*

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
BUUF	L*1	4-000002	004106	1059 (2118)
BUF	I*2	F-000006*	020000	4096 (1024,4)
BUJRD	L*1	4-000000	000002	1 (2)
I*2	I*2	4-000002	004106	1059 (1059)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
50	**	100	**				

TOTAL SPACE ALLOCATED = 004526 1193

NO FPP INSTRUCTIONS GENERATED

IT117: [320, 401A D/-SP-SY0: [320, 40]PATCRD

# OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14.35.54	30-OCT-80	PAGE 1
PATCHD.FTN		/TR: BLOCKS/WR		
0001		SUBROUTINE PATCHD(LUN, IPATCH, BUF)		
	C			
	C	THIS SUBROUTINE WILL READ ONE CHANNEL OF ONE PATCH IMAGE		
	C			
	C	CALL ARGUMENTS:		
	C	LUN	I	LOGICAL UNIT NUMBER TO READ PATCH INFO
	C	IPATCH	I	PATCH NUMBER REQUESTED
	C	BUF	O	BUFFER FOR DATA FROM PATCH
	C			
0002		BYTE BBUF(2118), BWORD(2)		
0003		INTEGER IBUF(1059), BUF(1024, 4)		
0004		EQUIVALENCE (IBUF, BBUF), (IWORD, BWORD)		
0005		IWORD=0		
	C			
	C	CALCULATE RECORD NUMBER FOR FIRST LINE OF PATCH DATA		
	C	ILIN=LINE NUMBER OF IMAGE WHERE PATCH DATA BEGINS		
	C	IPIX=PIXEL NUMBER FOR START OF PATCH DATA IN ILIN		
	C			
0006		ILIN=(IPATCH-1)/16*32+1		
0007		IREC=ILIN		
0008		INDX1=1		
0009		IPIX=MOD((IPATCH-1), 16)*32+70		
	C			
	C	DO LOOP TO READ 32 LINES OF PATCH		
	C			
0010		DO 100 I=1, 32		
0011		READ(LUN, IREC) IBUF		
	C			
	C	DO LOOP TO MOVE 32 PIXELS TO OUTPUT BUFFER CONVERTING TO I*2		
	C			
0012		DO 50 J=1, 32		
0013		BWORD(1)=BBUF(IPIX+J)		
0014		BUF(INDX1, 1)=IWORD		
0015		BWORD(1)=BBUF(IPIX+J+512)		
0016		BUF(INDX1, 2)=IWORD		
0017		BWORD(1)=BBUF(IPIX+J+1024)		
0018		BUF(INDX1, 3)=IWORD		
0019		BWORD(1)=BBUF(IPIX+J+1536)		
0020		BUF(INDX1, 4)=IWORD		
0021		INDX1=INDX1+1		
0022	50	CONTINUE		
0023		IREC=IREC+1		
0024	100	CONTINUE		
0025		RETURN		
0026		END		

ORIGINAL FILE  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:36:22	30-OCT-80	PAGE 1
VIN.FTN		/TR:BLOCKS/WR		
C VIN.FTN				
0001	SUBROUTINE VIN(ID,N)			
0002	IMPLICIT INTEGER (A-Z)			
	C			
	C	1	GREENNESS	
	C	2	GREEN NUMBER	
	C	3	ASHBURN VEGETATIVE INDEX	
	C	4	PERPENDICULAR VEGETATION INDEX	
	C	5	TRANSFORMED VEGETATION INDEX 6 TVI6	
	C	6	TRANSFORMED VEGETATION INDEX 7 TVI7	
	C	7	LEAF AREA INDEX SIMPLE FORM	
	C	8	LEAF AREA INDEX COMPLEX FORM	
	C	9	BEREKLY 2*CH4/CH2	
	C	10	DVI	
	C	11	RVI	
	C	12	UNUSED	
	C			
0003	COMMON/PATCH/CH, KH, FLAG, X, GAR, CLOUD, WATER, GAM, VEG, GIN, IAG,			
	1 SAT, SLN, GAMA			
0004	INTEGER*2 IAG, SAT, CH(1024,4), KH(1024,4)			
0005	INTEGER*2 GAR, CLOUD, WATER, GIN(3), SLN			
0006	BYTE FLAG(1024)			
0007	REAL GAM, X(4), VEG(12), GAMA			
0008	REAL VEGN			
	C			
0009	VEG(ID)=0			
0010	ICNT=0			
0011	IF(ID.LT.1 .OR. ID.GT.12) GO TO 333			
0012	GO TO (1,2,3,4,5,6,7,8,9,10,11,12) ID			
	C			
0013	1	CONTINUE		
0014	DO 15 I=1,N			
0015	IF(FLAG(I).NE.1)GO TO 15			
0016	VEGN=-.283*CH(I,1)-.660*CH(I,2)+.577*CH(I,3)+.388*CH(I,4)			
0017	VEG(1)=VEG(1)+VEGN			
0018	ICNT=ICNT+1			
0019	15	CONTINUE		
0020	GO TO 200			
0021	2	CONTINUE		
0022	DO 20 I=1,N			
0023	IF(FLAG(I).NE.1)GO TO 20			
0024	VEGN=-.283*CH(I,1)-.660*CH(I,2)+.577*CH(I,3)+.388*CH(I,4)			
0025	VEGN=VEGN+5			
0026	IF(VEGN.LT.0)VEGN=0			
0027	VEG(2)=VEG(2)+VEGN			
0028	ICNT=ICNT+1			
0029	20	CONTINUE		
0030	GO TO 200			
12 0031	3	CONTINUE		
11 0032	DO 30 I=1,N			
10 0033	IF(FLAG(I).NE.1)GO TO 30			
9 0034	VEGN=2*CH(I,4)-CH(I,2)			
8 0035	IF(VEGN.LT.0) VEGN=0.			
7 0036	VEG(3)=VEG(3)+VEGN			
6 0037	ICNT=ICNT+1			
5 0038	30	CONTINUE		

ORIGINAL FILE  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14 36 22	30-OCT-80	PAGE 2
VIN. FTN		/TR. BLOCKS/WR		
0039		GO TO 200		
0040	4	CONTINUE		
0041		DO 40 I=1,N		
0042		IF (FLAG(I).NE.1)GO TO 40		
0043		VEGN=ABS(.38425*(2.4*CH(I,4)-CH(I,2)))		
0044		IF (VEGN.LT.0)VEGN=0		
0045		VEG(4)=VEG(4)+VEGN		
0046		ICNT=ICNT+1		
0047	40	CONTINUE		
0048		GO TO 200		
0049	5	CONTINUE		
0050		DO 50 I=1,N		
0051		IF (FLAG(I).NE.1)GO TO 50		
0052		VEGN=SQRT(ABS((CH(I,3)-CH(I,2))/(CH(I,3)+CH(I,2))+0.5))		
0053		VEG(5)=VEG(5)+VEGN		
0054		ICNT=ICNT+1		
0055	50	CONTINUE		
0056		GO TO 200		
0057	6	CONTINUE		
0058		DO 60 I=1,N		
0059		IF (FLAG(I).NE.1)GO TO 60		
0060		VEGN=SQRT(ABS((2*CH(I,4)-CH(I,2))/(2*CH(I,4)+CH(I,2))+0.5))		
0061		IF ((2*CH(I,4)-CH(I,2)).LT.0)VEGN=0		
0062		VEG(6)=VEG(6)+VEGN		
0063		ICNT=ICNT+1		
0064	60	CONTINUE		
0065		GO TO 200		
0066	7	CONTINUE		
0067		DO 70 I=1,N		
0068		IF (FLAG(I).NE.1)GO TO 70		
0069		VEGN=1.653*CH(I,1)/CH(I,2)-1.698*CH(I,1)/CH(I,3)+.093		
0070		VEG(7)=VEG(7)+VEGN		
0071		ICNT=ICNT+1		
0072	70	CONTINUE		
0073		GO TO 200		
0074	8	CONTINUE		
0075		DO 80 I=1,N		
0076		IF (FLAG(I).NE.1)GO TO 80		
0077		VEGN=2.677 - 3.694*CH(I,1)/CH(I,2)		
	*	- 2.309*CH(I,1)/CH(I,3)		
	*	+ 2.375*CH(I,1)/CH(I,4)		
	*	+ 0.043*CH(I,2)/CH(I,3)		
	*	- 1.346*CH(I,2)/CH(I,4)		
	*	+3.017*((CH(I,1)/CH(I,2))-(CH(I,1)/(2*CH(I,4))))		
	*	*CH(I,1)/CH(I,2)		
0078		VEG(8)=VEG(8)+VEGN		
0079		ICNT=ICNT+1		
0080	80	CONTINUE		
12 0081		GO TO 200		
11 0082	9	CONTINUE		
10 0083		DO 90 I=1,N		
9 0084		IF (FLAG(I).NE.1)GO TO 90		
8 0085		VEGN=2*CH(I,4)/CH(I,2)		
7 0086		VEG(9)=VEG(9)+VEGN		
6 0087		ICNT=ICNT+1		
5 0088	90	CONTINUE		

ORIGINAL FILE  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:36:22	30-OCT-80	PAGE 3
VIN. FTN		/TR: BLOCKS/WR		
0089		GO TO 200		
0090	10	CONTINUE		
0091		DO 100 I=1,N		
0092		IF(FLAG(I).NE.1)GO TO 100		
0093		VEGN=CH(I,2)-2.4*CH(I,4)		
0094		VEG(10)=VEG(10)+VEGN		
0095		ICNT=ICNT+1		
0096	100	CONTINUE		
0097		GO TO 200		
0098	11	CONTINUE		
0099		DO 110 I=1,N		
0100		IF(FLAG(I).NE.1)GO TO 110		
0101		VEGN=CH(I,2)/CH(I,4)		
0102		VEG(11)=VEG(11)+VEGN		
0103		ICNT=ICNT+1		
0104	110	CONTINUE		
0105		GO TO 200		
0106	12	CONTINUE		
0107		GO TO 200		
0108	333	WRITE(6,334) ID		
0109	334	FORMAT(' BAD ID =',I8,' SUBROUTINE VIN')		
0110		VEG(ID)=9999.		
0111		RETURN		
0112	200	IF(ICNT.NE.0)VEG(ID)=VEG(ID)/ICNT		
0113		IF(ICNT.EQ.0)VEG(ID)=0		
0114		RETURN		
0115		END		

FORTRAN IV-PLUS V02-51E 14:36:22 30-OCT-80 PAGE 4  
VIN FTN / TR BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	002736 751	RW, I, CON, LCL
2	\$PDATA	000132 45	RW, D, CON, LCL
3	\$IDATA	000050 20	RW, D, CON, LCL
4	\$VARS	000010 4	RW, D, CON, LCL
5	\$TEMPS	000032 13	RW, D, CON, LCL
6	PATCH	042132 8749	RW, D, DVR, CBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
VIN		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CLOUD	I*2	6-042022	CAM	R*4	6-042026	GAMA	R*4	6-042126
TAG	I*2	6-042120	ICNT	I*2	4-000004	ID	I*2	F-000002*
SLN	I*2	6-042124	VEGN	R*4	4-000000	WATER	I*2	6-C42024

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
CH	I*2	6-000000	020000	4096 (1024, 4)
FLAG	L*1	6-040000	002000	512 (1024, 1)
GIN	I*2	6-042112	000006	3 (3)
WH	I*2	5-020000	020000	4096 (1024, 4)
VEC	R*4	6-042032	000060	24 (12)
X	R*4	6-042000	000020	8 (4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000102	2	1-000272	3	1-000476	4	1-000644
5	1-001206	7	1-001412	8	1-001574	9	1-002120
11	1-002414	12	1-002550	15	1-002444	20	1-000450
30	1-000774	50	1-001160	60	1-001364	70	1-001546
90	1-002232	100	1-002370	110	1-002524	200	1-002642
334	3-000000					333	1-002560

FUNCTIONS AND SUBROUTINES REFERENCED

\$SORT

FORTRAN IV-PLUS V02-51E

14:36:22

30-OCT-80

PAGE 5

VIN.FTN

/TR BLOCKS/WR

TOTAL SPACE ALLOCATED = 045334 9582

, TT17: (320, 40)A. B/-SP=SY0: (320, 40)VIN

ORIGINAL FILE  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14 36 42      30-OCT-80      PAGE 1
GAMMA. FTN      /TR. BLOCKS/WR

      C  GAMMA. FTN
      C
0001      SUBROUTINE GAMMA
0002      COMMON/PATCH/CH, KH, FLAG, X, GAR, CLOUD, WATER, GAM, VEG, GIN, IAG,
      1 SAT, SLN, GAMA
0003      INTEGER*2 IAG, SAT, CH(1024, 4), KH(1024, 4)
0004      INTEGER*2 GAR, CLOUD, WATER, GIN(3), SLN
0005      BYTE FLAG(1024)
0006      REAL GAM, X(4), VEG(12), GAMA
      C  GAMMA COMPUTES THE XSTAR OPTICAL DEPTH PARAMETER GAMMA
      C  IAG IS SUN ELEVATION ANGLE IN DEGREES
      C  GMA IS THE HAZE PARAMETER OUTPUT
0007      REAL AA, BB, CC, DD, XX(4), Y(5), Z(4), ANG, S
      C .....
      C .....
      D      WRITE(6, 16) IAG, SAT
0008      16      FORMAT('      GAMA      IAG=', I5, '      SAT=', I5)
0009      IF(IAG .LT. 15 .OR. IAG .GT. 75) GO TO 333
0010      IF(SAT .LT. -2 .OR. SAT .GT. 3) GO TO 333
0011      1      ANG=IAG *3.14159 /180.0
0012      CR=.77713/SIN(ANG)
0013      LK=SAT
0014      CALL CALIB(LK, 1024)
0015      CALL SCRALL(1024)
0016      CALL MEAN(1, 1024, NN, S)
0017      CALL MEAN(2, 1024, NN, S)
0018      CALL MEAN(3, 1024, NN, S)
0019      CALL MEAN(4, 1024, NN, S)
0020      GO TO 567
0021      ENTRY KGMA(Y)
0022      X(1)=Y(1)
0023      X(2)=Y(2)
0024      X(3)=Y(3)
0025      X(4)=Y(4)
0026      CR =Y(5)
0027      567      CONTINUE
      D      WRITE(6, 17) X
0028      17      FORMAT('      GAMMA MEANS      ', 4F8.3)
0029      Z(1)=CR*X(1)
0030      Z(2)=CR*X(2)
0031      Z(3)=CR*X(3)
0032      Z(4)=CR*X(4)
0033      AA=1.6078*(Z(1)-61.9)*(-0.8995)
0034      AA=AA+1.0910*(Z(2)-66.2)*0.4283
0035      AA=AA+0.8358*(Z(3)-83.2)*0.0759
0036      AA=AA-0.5981*(Z(4)-33.9)*0.0408
0037      BB=-1.2680*(Z(1)-61.9)*0.8995
0038      BB=BB+1.0445*(Z(2)-66.2)*0.4283
12 0039      BB=BB+0.9142*(Z(3)-83.2)*0.0759
11 0040      BB=BB-0.7734*(Z(4)-33.9)*0.0408
10 0041      CC=-Z(1)*0.8995+Z(2)*0.4283+Z(3)*0.0759-Z(4)*0.0408+11.2082
9 0042      DD=BB*BB-2.0*AA*CC
8 0043      IF(DD .LT. 0.0) DD=0.0
7 0044      GAMA=(BB-SQRT(DD))/AA
6      D      WRITE(6, 2020) GAMA
5 0045      GAM=GAMA*1000+300
4
3

```



ORIGINAL FROM  
OF FOUR COPIES

FORTAN IV-PLUS V02-51E

14:36:42

30-OCT-80

PAGE 2

GAMMA, FTN	/TR: BLOCKS/WR
------------	----------------

0046 2020 FORMAT(//, ' GAMMA =', F8.4, //)

0047	RETURN
------	--------

0048	333	TYPE 8, IAG, SAT, NN
------	-----	----------------------

```
0049      8      FORMAT( '      PROBLEMS      IAG= ', I6, '      SAT= ', I6, '      NN= ', I6,
```

```
*      /, '      S TO STOP      CR TO CONTINUE', 10X, 'GAMMA')
```

0050 ACCEPT 9, 11TT

```
0051          9      FORMAT(A1)
```

```
0052      IF(IITT.EQ.'S') STOP
```

0053 GO TO 1

0054 END

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14 36.42 30-OCT-80 PAGE 3  
GAMMA.FTN /TR.BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	001266	347 RW, I, CON, LCL
2	\$DATA	000144	50 RW, D, CON, LCL
3	\$IDATA	000250	76 RW, D, CON, LCL
4	\$VARS	000102	33 RW, D, CON, LCL
5	\$TEMPS	000030	12 RW, D, CON, LCL
6	PATCH	042132	8749 RW, D, OVR, GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
GAMMA		1-000000	KCMA		1-000236						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
AA	R*4	4-000000	ARG	R*4	4-000060	BB	R*4	4-000004	CC	R*4	4-000010
CH	R*4	4-000070	DD	R*4	4-000014	GAM	R*4	4-042026	GAMA	R*4	6-042126
GIN	I*2	6-042120	IIIT	I*2	4-000100	LK	I*2	4-000074	NN	I*2	4-000076
SAT	I*2	6-042122	SLN	I*2	6-042124	WATER	I*2	6-042024	S	R*4	4-000064

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
CH	I*2	6-000000	020000	4096 (1024,4)
FLAG	I*1	6-040000	062000	512 (1024)
GIN	I*2	6-042112	000006	3 (3)
KH	I*2	6-020000	020000	4096 (1024,4)
VEC	R*4	6-042032	000060	24 (12)
X	R*4	6-042000	000020	8 (4)
Y	R*4	4-000020	000020	8 (4)
Z	R*4	F-000002*	000024	10 (5)
	R*4	4-000040	000020	8 (4)

TAGELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000072	B'	3-000000	9'	3-000130	16'	**
333	1-001114	567	1-000334	2020'	**	17'	**

FUNCTIONS AND SUBROUTINES REFERENCED

CALIU	MEAN	SCRALL	\$SIN	\$SQRT
-------	------	--------	-------	--------

ORIGINAL PAGE 13  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E  
GAMMA.FTN /TR:BLOCKS/WR

14:36 42

30-OCT-80

PAGE 4

TOTAL SPACE ALLOCATED = 044146 9267

, TT17: (320, 40)A. B/-SP=SYO: (320, 40)GAMMA

ORIGINAL PAGE 17  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:37:30	30-OCT-80	PAGE 1
MEAN.FTN		/TR: BLOCKS/WR		
C	MEAN COMPUTES MEAN AND STANDARD DEVIATION FOR IX(I)			
0001	SUBROUTINE MEAN(ICHAN,N,NN,STDEV) 'MEAN FOR FLAG(I)=1			
0002	IMPLICIT INTEGER (A-Z)			
0003	COMMON/PATCH/CH,KH,FLAG,X,GAR,CLOUD,WATER,GIN, IAG,			
	1 SAT, SLN,GAMA			
0004	INTEGER*2 IAG,SAT,CH(1024,4),KH(1024,4)			
0005	INTEGER*2 GAR,CLOUD,WATER,GIN(3),SLN			
0006	BYTE FLAG(1024)			
0007	REAL GAM,X(4),VEG(12),GAMA			
0008	INTEGER*2 N,NN			
0009	REAL S,SS			
0010	REAL STDEV,VARI,XNN			
D	TYPE B0B			
0011	FORMAT(' MEAN')			
0012	X(ICHAN)=0.			
0013	S=0.			
0014	SS=0.			
0015	NN=0			
0016	DO 1 I=1,N			
0017	IF(FLAG(I).NE.1) GO TO 1			
0018	NN=NN+1			
0019	S=S+CH(I, ICHAN)			
0020	SS=SS+CH(I, ICHAN)**2			
0021	1 CONTINUE			
0022	IF(NN.EQ.0)RETURN			
0023	X(ICHAN)=S/NN			
0024	VARI=(SS-X(ICHAN)*S)/(NN-1)			
0025	STDEV=SQRT(ABS(VARI))			
D	TYPE B09,NN,X(ICHAN),STDEV			
0026	FORMAT(' ',I4,F8.2,F8.2)			
0027	RETURN			
0028	END			

FORTRAN IV-PLUS V02-51E 14.37.30 30-OCT-80 PAGE 2  
MEAN FIN /TR: BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000324	106 RW, I, CON, LCL
4	\$VARS	000022	9 RW, D, CON, LCL
5	\$TEMPS	000002	1 RW, D, CON, LCL
6	PATCH	042132	8749 RW, D, OVR, GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
MEAN		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
FLAG	I*2	6-042022	GAM	R*4	6-042026	GAMA	R*4	6-042126
ICHAN	I*2	6-042120	ICHAN	I*2	6-000002	NN	I*2	6-000004
SLN	I*2	6-042122	SLN	I*2	6-042124	SS	R*4	4-000004
XNN	R*4	6-042024	XNN	R*4	4-000014	STDEV	R*4	F-000010
						VARI	R*4	4-000010

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
CH	I*2	6-000000	020000	4096 (1024,4)
FLAG	I*1	6-040000	000000	512 (1024)
GIN	I*2	6-042112	000006	3 (3)
WH	I*2	6-020000	020000	2096 (1024,4)
VEG	R*4	6-042032	000060	24 (12)
X	R*4	6-042000	000020	8 (4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000172	808'	**	809'	**

FUNCTIONS AND SUBROUTINES REFERENCED

\$SGRT	
TOTAL SPACE ALLOCATED = 042502	8865
TT17 (320.40)A B/-SP=SYO (320.40)MEAN	

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14 37 46	30-OCT-80	PAGE 1
KAUTH FTN		/TR BLOCKS/WR		
C KAUTH COMPUTES THE KAUTH THOMAS TRANSFORMATION				
0001		SUBROUTINE KAUTH(N)		
0002		IMPLICIT INTEGER*2 (A-S)		
0003		COMMON/PATCH/CH,KH,FLAG,X,GAR,CLOUD,WATER,GAM,VEG,GIN,IAG,		
		I SAT,SLN,GAMA		
0004		INTEGER*2 IAG,SAT,CH(1024,4),KH(1024,4)		
0005		INTEGER*2 GAR,CLOUD,WATER,GIN(3),SLN		
0006		BYTE FLAG(1024)		
0007		REAL GAM,X(4),VEG(12),GAMA		
0008		REAL KHC(4)		
0009		REAL R(4,4)		
0010		DATA R / 332, 603, 676, 263,		
	C	- 233, - 660, 577, 388,		
	C	- 900, 429, 076, - 040,		
	C	- 016, 131, - 452, 882/		
	D	WRITE(6,66) R		
0011	66	FORMAT(' KAUTH MATRIX',/,4(' ',4F6.4,/))		
0012		DO 100 I=1,N		
0013		DO 101 J=1,4		
0014		KHC(J)=30 +CH(I,1)*R(1,J)+CH(I,2)*R(2,J)+CH(I,3)*R(3,J)+		
		I CH(I,4)*R(4,J)		
0015	101	CONTINUE		
0016		KH(I,1)=IIFIX(KHC(1)-29.5)		
0017		KH(I,2)=IIFIX(KHC(2)-29.5)		
0018		IF(KH(I,2) .LT. 0) KH(I,2)=KH(I,2)-1		
0019		KH(I,3)=IIFIX(KHC(3)-29.5)		
0020		IF(KH(I,3) .LT. 0) KH(I,3)=KH(I,3)-1		
0021		KH(I,4)=IIFIX(KHC(4)-29.5)		
0022	100	CONTINUE		
0023		RETURN		
0024		END		

ORIGINAL PAGE  
OF POOR QUALITY

FORTTRAN IV-PLUS V02-51E 14:37:46 30-OCT-80 PAGE 2  
KAUTH.FTN /TR BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000322 105	RM.1.COM.LCL
4	\$VARS	000124 42	RM.D.COM.LCL
5	\$TEMPS	000012 5	RM.D.COM.LCL
6	PATCH	042132 8749	RM.D.DVR.GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
KAUTH		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CLOUD	I*2	6-042032	CAM	R*4	6-042026	GAR	I*2	6-042020
IAG	I*2	6-042120	J	I*2	4-000122	SAT	I*2	6-042122
WATER	I*2	6-042024						

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
CH	I*2	6-000000	020000	4096 (1024,4)
FLAG	L*1	6-040000	002000	512 (1024)
GIN	I*2	6-042112	000006	3 (3)
KH	I*2	6-020000	020000	4096 (1024,4)
KHC	R*4	4-000000	000020	8 (4)
R	R*4	4-000020	000100	32 (4,4)
VEG	R*4	6-042032	000060	24 (12)
X	R*4	6-042000	000020	8 (4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
66'	**	100	**	101	**

TOTAL SPACE ALLOCATED = 042612 8901

.1117.1320.401A.B/-SP-SVD.1320.401KAUTH

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14.40.07 30-OCT-80 PAGE 1  
SCRALL.FTN /TR:BLOCKS/WR

C SUBROUTINE SCRALL KAUTH TRANSFORMS AND SCREENS CH DATA  
C  
C COMPILE WITH THE CO:50 SWITCH.  
C

```

0001      SUBROUTINE SCRALL(N)
0002      IMPLICIT INTEGER (A-Z)
0003      COMMON/PATCH/CH,KH,FLAG,X,GAR,CLOUD,WATER,GAM,VEG,GIN,IAG,
          1 SAT,SLN,GAMA
0004      INTEGER*2 IAG,SAT,CH(1024,4),KH(1024,4)
0005      INTEGER*2 GAR,CLOUD,WATER,GIN(3),SLN
0006      BYTE FLAG(1024)
0007      REAL GAM,X(4),VEG(12),GAMA
0008      REAL R(4,4),ZZ,C,Z(4,19),RR(4,4),Y(4),CC(19)
0009      INTEGER II(19),JJ(17),LL(21),XX(4),T
0010      INTEGER*4 I4
0011      INTEGER*2 SSN
0012      DATA RR/.33231,.60316,.67581,.26278,
          C      -.28317,-.66006,.57735,.38833,
          C      -.89952,.42830,.07592,-.04080,
          C      -.01544,.13068,-.45187,.88232/
0013      DATA Z/0.0,0.0,0.0,0.0,1.0,
          C      0.0,0.0,0.0,-1.0,
          C      -.09375,0.0,1.0,0.0,
          C      -.1875,0.0,-1.0,0.0,
          C      -0.1,-1.0,0.0,0.0,
          C      0.555556,1.0,0.0,0.0,
          C      -0.83333333,1.0,0.0,0.0,
          C      1.0,0.0,0.0,0.0,
          C      -.1,0.0,-1.0,0.0,
          C      1.0,0.0,0.0,0.0,
          C      -0.142857,0.0,-1.0,0.0,
          C      -1.0,0.0,0.0,0.0,
          C      -.0675,-1.0,0.0,0.0,
          C      0.0,0.0,0.0,-1.0,
          C      0.0,-1.0,0.0,-1.0,
          C      -.5,-1.0,-1.0,-0.5,
          C      -.4,1.0,0.0,0.0,
          C      -0.4,1.0,-0.6,-0.6,
          C      -1.0,0.4,0.0,0.0/
0014      DATA CC/16.,12.,-4.,14.,20.,156.,-8.,100.,7.5,69.,3.25,
          C      -75.,0.5,-1.5,4.5,-10.,-12.2,-9.,-37.75/
0015      DATA II/2,3,4,5,6,7,8,10,10,12,12,18,18,18,18,18,40,20,20/
0016      DATA JJ/60,60,60,60,60,60,60,9,60,11,60,13,14,15,16,17,21,
          C      19,60/
0017      DATA LL/47,46,45,44,43,42,41,34,33,32,31,26,25,24,23,22,21,
          *      13,12,1,11/
          C      4* GARBLED; 34,33 CLOUD; 32,31 HAZE
          C      2* WATER; 1* SHADOW, 11 OVER WATER; 1 GOOD.
12      C
11      D      TYPE B08
10      0018      B08      FORMAT(' SCRALL')
9      0019      SSN=30
8      0020      GAR=0
7      0021      CLOUD=0
6      0022      WATER=0
5      0023      GIN(1)=0

```



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:40:07	30-OCT-80	PAGE 2
SCRALL.FTN		/TR: BLOCKS/WR		
0024		GIN(2)=0		
0025		GIN(3)=0		
0026		ICNT=0		
0027	600	IF(IAQ.LT.15.OR. IAQ.GT.80)TYPE 101,SEQ.DATE		
0028	101	FORMAT(' INPUT SUNANGLE FOR',I6.2I3,' SCRALL')		
0029		IF(IAQ.LT.15.OR. IAQ.GT.80) ACCEPT 201,IAQ		
0030	201	FORMAT(I2)		
0031		ZZ=IAQ/57.3		
0032		IF(ZZ.LT.0.1) ZZ=0.907		
0033		C=0.77713/SIN(ZZ)		
	D	TYPE 990,IAQ,C		
	D	WRITE(8,990) IAQ,C		
0034		IF (C.LT.0.) TYPE 990,IAQ,C		
0035		IF (C.LT.0.) RETURN		
0036	990	FORMAT(' C ',I4,F10.4)		
0037		DO 100 K=1,N		
0038		DO 5 I=1,4		
0039	5	Y(I)=0		
0040		XX(1)=CH(K,1)		
0041		XX(2)=CH(K,2)		
0042		XX(3)=CH(K,3)		
0043		XX(4)=CH(K,4)		
	D	IF(K.EQ.1) TYPE 995,X		
0044	995	FORMAT(' X ',4I5)		
0045		DO 6 I=1,4		
0046		DO 6 J=1,4		
0047	6	Y(I)=Y(I)+XX(J)*RR(J,I)		
0048		ISN=Y(2)+.5		
0049		IF(ISN.LT.SSN)SSN=ISN		
0050	98	DO 66 I=1,4		
0051	66	Y(I)=Y(I)*C ! SUN ANGLE CORRECTION		
0052		L=1		
0053	1	CONTINUE		
	D	IF(K.LT.3) TYPE 996,L,T,U,CC(L)		
	D	IF(K.LT.20) WRITE(8,996) L,T,U,CC(L),Y		
0054	996	FORMAT(' ',I4,I4,5F10.4,F10.4)		
0055		IF(L.LE.21)T=LL(L)		
0056		IF(L.GE. 20) GO TO 99		
0057		U=0.		
0058		DO 7 I=1,4		
0059	7	U=U+Y(I)*Z(I,L)		
0060		IF(U.LT. CC(L)) GO TO 3		
0061		L=JJ(L)		
0062		GO TO 1		
0063	3	CONTINUE		
0064		L=II(L)		
0065		GO TO 1		
0066	99	FLAG(K)=T		
12 0067		IF(T.EQ.1)ICNT=ICNT+1		
11 0068		IF(T.GE.40.AND.T.LE.49)GAR=GAR+1		
10 0069		IF(T.GE.30.AND.T.LE.39)CLOUD=CLOUD+1		
9 0070		IF(T.GE.10.AND.T.LE.29)WATER=WATER+1		
8 0071		IF(T.NE.1)GO TO 100		
7 0072		IF(Y(2).GE.25.)GIN(3)=GIN(3)+1		
6 0073		IF(Y(2).GE.10.)GIN(2)=GIN(2)+1		
5 0074		IF(Y(2).GE.0)GIN(1)=GIN(1)+1		

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E          14:40:07          30-OCT-80          PAGE 3
SCRAL L. FTN          /TR: BLOCKS,WR

0075      100      CONTINUE
0076              SLN=SSN
0077              IF(SLN.LT.-10)SLN=-10
0078              IF(ICNT.LE.0)RETURN
0079              I4=GIN(1)
0080              I4=I4*100
0081              GIN(1)=I4/ICNT
0082              I4=GIN(2)
0083              I4=I4*100
0084              GIN(2)=I4/ICNT
0085              I4=GIN(3)
0086              I4=I4*100
0087              GIN(3)=I4/ICNT
              D      TYPE 991,FLAG
0088      991              FORMAT(' FLAG',19I3)

0089              RETURN
0090              END

```

ORIGINAL PAGE  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:40:07 30-OCT-80 PAGE 4  
SCROLL FTR 7TR BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	SCODE1	001670 476	RW, I, CON, LCL
2	SPDATA	000020 8	R, D, CON, LCL
3	SDATA	000074 30	RW, D, CON, LCL
4	SVARS	001254 342	RW, D, CON, LCL
5	FIEMPS	000006 3	RW, D, CON, LCL
6	PATCH	042132 8749	RW, D, DNR, GDL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
SCROLL		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
C	R+4	4-000104	CLOUD	I+2	6-043022	DATE	I+2	4-001236
GAR	I+2	6-045020	I	I+2	4-001242	IAG	I+2	6-042120
IA	I+4	4-001244	J	I+2	4-001244	K	I+2	4-001240
SAT	I+2	6-042122	SEC	I+2	4-001234	SLN	I+2	6-042124
U	I+2	4-001252	WATER	I+2	6-042024	ZZ	R+4	4-000100

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
CC	R+4	4-000710	000114	38 (19)
CH	I+2	6-000000	020000	4096 (1024,4)
FLAG	I+1	6-040000	002000	512 (1024)
GIN	I+2	6-042112	000006	3 (3)
II	I+2	4-001024	000046	19 (19)
JJ	I+2	4-001072	000046	19 (19)
KH	I+2	6-020000	020000	4096 (1024,4)
LL	I+2	4-001140	000052	21 (21)
R	R+4	4-000300	000100	32 (4,4)
RR	R+4	4-000570	000100	32 (4,4)
VEG	R+4	6-042052	000060	24 (12)
X	R+4	6-042000	000020	8 (4)
XX	I+2	4-001252	000010	4 (4)
Y	R+4	4-000670	000020	8 (4)
Z	R+4	4-000110	000460	152 (4,19)

LABELS

LABEL	ADDR:SS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000722	3	1-001122	5	**	6	**
						7	**

FORTRAN IV-PLUS V02-51E 14: 40: 07 30-OCT-80 PAGE 5  
SCROLL.FIN /TR: BLOCKS/WR

66	**	98	**	99	1-001146	100	1-001340	101	3-000000
201	3-000052	600	**	808	**	990	3-000056	991	**
995	**	996	**						**

## FUNCTIONS AND SUBROUTINES REFERENCED

**NIS**

TOTAL SPACE ALLOCATED = 045420 9608

.TT17·[320,40]A.B/-SP=SY0:[320,40]SCROLL

FORTRAN IV-PLUS V02-51E 14:40:20 30-OCT-80 PAGE 1  
CALIB.FTN /TR: BLOCKS/LR

```

C CALIB APPLIES CALIBRATION CONSTANTS TO DOT DATA
C LK=-1 FOR POSTLAUNCH LANDSAT 2
C LK= 0 FOR NO CORRECTION
C LK= 1 FOR LANDSAT 1 CALIBRATION
C LK= 2 FOR PRELAUNCH LANDSAT 2 (NO CORRECTION)
C LK= 3 FOR LEC LANDSAT 3 CALIBRATION
C LK= 4 FOR ERM LANDSAT 3 CALIBRATION
C SUBROUTINE CALIB(LK,N)
C IMPLICIT INTEGER*2 (A-S)
C COMMON/PATCH/CH,KH,FLAG,X,GAR,CLOUD,WATER,GAM,VEG,GIN,IAG,
C 1 SAT,SLN,GAMA
C INTEGER*2 IAG,SAT,CH(1024,4),KH(1024,4)
C INTEGER*2 GAR,CLOUD,WATER,GIN(3),SLN
C BYTE FLAG(1024)
C REAL GAM,X(4),VEG(12),GAMA
C REAL CALFAC(4,4),CALBAS(4,4)
C DATA CALFAC/1.040,1.000,1.000,1.090,0.820,
C 1.275,1.141,1.098,0.948,
C 1.161,1.230,1.246,1.062,
C 1.137,1.173,1.247,1.126/
C DATA CALBAS/-5.790,1.190,-2.910,3.010,
C -1.445,-2.712,-2.950,0.446,
C 0.0,0.0,0.0,0.0,0.0,
C 0.0,0.0,0.0,0.0,0.0 /
D TYPE 1,LK
D TYPE 2,CALBAS(1,LK),CALBAS(2,LK),CALFAC(1,LK),CALFAC(2,LK)
0011 2 FORMAT(' CALIB B,A CHECK',4F8.3)
0012 1 FORMAT(' CALIB SAT=',I4)
0013 IF(LK.EQ.2) RETURN ! LACIE PRE-LAUNCH CALIBRATION
0014 IF(LK.EQ.0) RETURN ! 0 MEANS SATELLITE ID NOT FOUND
0015 IF(LK.EQ.-1) LK=2 ! -1 MEANS LANDSAT 2 POST LAUNCH CALIBRATION
0016 IF(LK.LT.1 .OR. LK.GT.4) WRITE(6,1) LK
0017 IF(LK.LT.1 .OR. LK.GT.4) RETURN
0018 DO 100 I=1,N
0019 CH(I,1)=0.5*CH(I,1)*CALFAC(1,LK)+CALBAS(1,LK)
0020 CH(I,2)=0.5*CH(I,2)*CALFAC(2,LK)+CALBAS(2,LK)
0021 CH(I,3)=0.5*CH(I,3)*CALFAC(3,LK)+CALBAS(3,LK)
0022 CH(I,4)=0.5*CH(I,4)*CALFAC(4,LK)+CALBAS(4,LK)
0023 100 CONTINUE
0024 RETURN
0025 END

```

ORIGINAL  
OF PAGE

FORTRAN IV PLUS V02-S1E 14:40:20 30-OCT-80 PAGE 2  
CALIB FTN /TR BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CORE1	000342	113 RM. I. CON. LCL
3	\$IDATA	000024	10 RM. D. CON. LCL
4	\$VARS	000202	65 RM. D. CON. LCL
5	\$TEMPS	000002	1 RM. D. CON. LCL
6	PATCH	042132	8749 RM. D. DVR. GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CALIB		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CLC00	1-2	6-042032	GAM	R-4	6-042024	GATA	R-4	6-042030
IAG	1-2	6-042120	LK	1-2	F-000002	N	1-2	6-042122
WATER	1-2	6-042024				SAT	1-2	6-042124

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
CALJAS	R-4	4-000100	000100	32 (4,4)
CALFAC	R-4	4-000000	000100	32 (4,4)
CH	1-2	6-000000	020000	4096 (1024,4)
FLAG	L-1	6-040000	002000	512 (1024)
GIN	1-2	6-042112	000006	3 (3)
WH	1-2	6-020000	020000	4096 (1024,4)
VEG	R-4	6-042032	000040	24 (12)
X	R-4	6-042000	000020	6 (4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1'	3-000000	2'	**	100	**		

TOTAL SPACE ALLOCATED = 042724 8938

TT17 (320.40)A B/-SP=SV0: (320.40)CALIB

## APPENDIX B

### LISTING OF SKIP PROCESSOR

ORIGINAL PAGE IS  
OF POOR QUALITY.

```

C THIS PROGRAM WILL READ A TAPE IN LANDSAT(1,2), UNIVERSAL,
C AND LARSYS FORMAT. WHILE THE TAPE IS READ, THE DATA IS
C SELECTED IN A SKIPPED PIXEL AND LINE PATTERN TO FORM A
C 512 * 512 IMAGE FROM THE FULL SCENE
C
C LOCAL VARIABLES:
C
C BFSZ= BUFFER SIZE IN BYTES
C BUF= IMAGE DATA BUFFER
C
0002      IMPLICIT INTEGER (A-Z)
0003      REAL FLAT, FLONG, NLAT, NLONG
0004      REAL H4, H5, H6, H7
0005      LOGICAL*1 BUFFER(3280), SCNID, BWORD(2), ANS,
+          FILEN(6), FLNIMD(16), FLNIMH(16), MISNO(2), BUF(850)
0006      DIMENSION IPRM(6), UIFHR(1536)
0007      COMMON/COMH/ST, SE, LS, LE, NOCHAN, NOSAMP, SCNID(12), SUNEL,
+          SUNAZ, SNSHD, NREV, EXYR, EXDAY, DAY, HR, MIN,
+          SEC, SEQNO(2), FLAT, FLONG, NLAT, NLONG
0008      COMMON/LABEL/NAME(3), MISNO, E
0009      COMMON/HIST/H4(256), H5(256), H6(256), H7(256)
C
0010      DATA UIFHR/1, 512, 1, 512, 1, 512, 1, 512, 3, 4, 1, 2, 1, 1,
+          1, 1, 4, 1, 2, 3, 196, 2, 16*0, 15861, -15729,
+          15861, -15729, 15861, -15729, 15861,
+          -15729, 15861, -15729, 15861, -15729,
+          15861, -15729, 15861, -15729, 1, 427, 1,
+          3*0, 4, 257, 257, 10*0, 512, 1059, 1, 2, 1459*0/
0011      DATA BFSZ/4000/WLB/'O400/, LUN1/1/, LUN2/2/, INIT/0/
0012      DATA FLNIMH/' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'I', 'M', 'H',
+          0, 0, 0, 0, 0, 0/
+          FLNIMD/' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', 'I', 'M', 'D',
+          0, 0, 0, 0, 0, 0/
C
C FORMAT LISTING:
C
0013      4 FORMAT(/, 1X, 10('*'), /, /* ERROR= ', 15, 'AT= ',
+          3A2, /, 1X, 10('*'))
0014      5 FORMAT(13)
0015      6 FORMAT(1X, 10('*'), /, /* REPLACE MOUNTED TAPE WITH NEXT ',
+          'TAPE IN SEQUENCE', /, 1X, 10('*'))
0016      7 FORMAT(1A1)
0017      CALL ASNLUN(3, 'TT', 0)
C
C INITIALIZING HISTOGRAMS TO ZERO
C
DO 10 L=1, 256
0018      DO 10 L=1, 256
0019      H4(L)=0
0020      H5(L)=0
0021      H6(L)=0
0022      H7(L)=0
C
0023      10 CONTINUE
C

```



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:30:53	30-OCT-80	PAGE 2
SKIP.FTN		/TR.BLOCKS/WR		
0024		READ(5,1000)FILEN		
0025	1000	FORMAT(6A1)		
	C			
	C	CREATING THE NAME OF THE FILES TO BE WRITTEN		
	C			
0026		DO 15 K=1,6		
0027		FLNIMH(K)=FILEN(K)		
0028		FLNIMD(K)=FILEN(K)		
0029	15	CONTINUE		
	C			
	C	READ START LINE AND START PIXEL		
	C			
0030		READ(5,1100)LINST		
0031	1100	FORMAT(1I2)		
0032		READ(5,1100)IPIXST		
	C			
	C	THIS ROUTINE WILL READ THE HEADER OF THE IMAGE TAPE		
	C			
0033		CALL GETADR(HDRADR,BUFFER)		
0034		CALL GIO("2400,LUN1)		
0035		CALL HDRREAD(LUN1,IFRM,BUFFER,BFSZ,EOF,PC,IE,BUFFER,INS,HDRADR,ISS)		
	C			
	C	CHECKING FOR ERRORS DURING TAPE READ		
	C	CHECKING FOR END OF FILE		
	C			
0036		IF(IE.NE.0)GOTO 40		
	C			
	C	WRITE THE HEADER IN IMDACS FORMAT		
	C			
0037		CALL HDRFLN(LUN2,FLNIMH,UIFHR,SUNEL,FLAT,FLONG)		
	C			
	C	OPEN THE IMAGE DATA FILE		
	C			
0038		OPEN(UNIT=LUN2,NAME=FLNIMD,TYPE='NEW',FORM='UNFORMATTED',		
		+ ACCESS='DIRECT',RECORDSIZE=640)		
	C			
	C	CALL SKPRD PROGRAM ONCE FOR EACH CCT STRIP		
	C			
0039		DO 100 ICCT=1,4		
0040		CALL SKPRD(LUN1,LUN2,BUF,BFSZ,EOF,IE,BUFFER,IFRM,HDRADR,ICCT,		
		1 LINST,IPIXST)		
0041		IF(IE.NE.0)GO TO 40		
0042		IF(ICCT.EQ.4)GO TO 50		
0043		IF(ICCT.NE.2)GO TO 60		
	C			
	C			
	C	INFORM THE OPERATOR TO MOUNT THE NEXT TAPE IF IT EXISTS		
	C			
12				
11	0044	CALL GIO("2400,LUN1)		
10	0045	WRITE(3,6)		
9	0046	35	READ(3,7)ANS	
8	0047	IF(ANS.NE.'C')GOTO 35		
7	0048	CALL GIO("2400,LUN1)		
6				
5				
4				
3				
2				
1				
0				
	C			
	C	READ HEADER RECORDS FOR NEXT STRIP		

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:30:53	30-OCT-80	PAGE 3
SKIP. PTN		/TR: BLOCKS/WR		
	C			
0049	40	CALL HDREAD(LUN1, IFRM, BUFFER, BFSIZE, EOF, PC, IE, BUFFER, INS, HDRADR, 1 ISS)		
0050		IF(IE.NE.0)GO TO 40		
0051	100	CONTINUE		
	C			
	C	ERROR		
	C			
0052	40	CONTINUE		
0053		WRITE(6,4)IE,NAME		
0054		GOTO 9999		
	C			
	C			
0055	50	CONTINUE		
	C			
	C	CLOSE THE IMAGE DATA FILE		
	C			
0056		CLOSE(UNIT=LUN2)		
0057		CALL QIO("2400,LUN1)		
	C			
	C	WRITE THE HISTOGRAMS TO THE HEADER FILE		
	C			
0058		CALL HDRHIS(LUN2,FLNIMH,UIFHR)		
	C			
0059	9999	STOP		
0060		END		

ORIGINAL PAGE IS  
OF POOR QUALITY

PAGE 4

30-OCT-80

14:30:53  
/TR:BLOCKS/MR

FORTRAN IV-PLUS V02-51E

SKIP FPN

PROGRAM SECTIONS

NUMER NAME SIZE ATTRIBUTES

1	SCODE1	000706	227	RM. I. COM. LCL
2	\$DATA	000050	8	RM. D. COM. LCL
3	\$DATA	000406	131	RM. D. COM. LCL
4	\$VARS	016172	3645	RM. D. COM. LCL
6	CMH	000100	32	RM. D. DMR. CEL
7	LABEL	000012	5	RM. D. DMR. CBL
8	MIST	010000	2048	RM. D. DMR. CBL

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ANS	L+1	4-006322	BFSZ	I+2	4-016130	DAY	I+2	6-000044
EDAY	I+2	6-000042	EXVR	I+2	6-000040	FLAT	R+4	6-000040
HR	I+2	6-000046	ICCT	I+2	4-016170	IE	I+2	4-016152
INS	I+2	4-016164	IPRST	I+2	4-016150	ISS	I+2	4-016142
LE	I+2	6-000006	LINST	I+2	4-016146	LS	I+2	4-016134
MIN	I+2	6-000050	MLAT	R+4	6-000070	NLONG	R+4	6-000074
NREV	I+2	6-000036	PC	I+2	4-016160	SE	I+2	6-000002
ST	I+2	6-000000	SUNAZ	I+2	6-000032	SUNEL	I+2	6-000030
								4-016132

ARRAYS

NAME TYPE ADDRESS SIZE DIMENSIONS

BUF	L+1	4-006371	001522	425	(850)
BUFFER	L+1	4-000000	006320	1640	(3260)
BLCRD	L+1	4-006320	000002	1	(2)
FILEN	L+1	4-006323	000006	3	(6)
FLNPD	L+1	4-006331	000020	8	(16)
FLNPDH	L+1	4-006351	000020	8	(16)
H4	R+4	8-000000	002000	512	(256)
H5	R+4	8-002000	002000	512	(256)
H6	R+4	8-004000	002000	512	(256)
H7	R+4	8-006000	002000	512	(256)
IPRM	I+2	4-010114	000014	6	(6)
MISRO	L+1	7-000006	000002	1	(2)
NAME	L+2	7-000000	000006	3	(3)
SCHD	L+1	6-000014	000014	6	(12)
SEPHD	I+2	6-000054	000004	2	(2)
UIFHR	I+2	4-010130	006000	1536	(1536)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
4	3-000000	5	3	6	6	7	7
15	1-000446	35	1-000466	40	1-000564	50	1-000524
100	3-000162	1000	3-000166	9999	1-000672	60	1-000524

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTAN IV-PLUS V02-51E 14:30:53 30-OCT-80 PAGE 5  
SKIP FTR /TR:BLOCKS/WR

FUNCTIONS AND SUBROUTINES REFERENCED

ASMLUN CLOS\$ GETADR HDREAD HDRFLN HDRHIS OPENS QIO SKPRD

TOTAL SPACE ALLOCATED = 027640 6096

, TT17: [320, 40]A. B/-SP=SYO: [320, 40]SKZP

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14 31 11 30-OCT-80 PAGE 1  
SKPRD. FTM /TR: BLOCKS/WR

0001 SUBROUTINE SKPRD(LUN1,LUN2,BUF,BFSZ,EOF,IE,BUFFER,IFRM,HDRADR,  
1 ICCT,LINST,IPIXST)

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

THIS SUBROUTINE WILL READ THE LINES OF DATA FROM THE LANDSAT TAPE  
AND SELECT THE PIXELS TO WRITE TO THE FILE.

CALL ARGUMENTS:

LUN1 LOGICAL UNIT OF TAPE DRIVE

LUN2 LOGICAL UNIT OF IMAGE DATA FILE

BUF BUFFER FOR DATA RETURNED FROM LDCODE

BFSZ SIZE OF BUF

EOF END OF FILE INDICATOR FROM LDCODE =1 FOR EOF

IE ERROR INDICATOR =1 FOR ERROR

BUFFER BUFFER FOR LDCODE TO READ TAPE RECORD INTO

IFRM FORMAT OF TAPE =2 FOR LANDSAT

HDRADR ADDRESS OF BUFFER

ICCT STRIP NUMBER CURRENTLY PROCESSING

LINST START LINE FOR SKIPPED IMAGE

IPIXST START PIXEL FOR SKIPPED IMAGE

0002 IMPLICIT INTEGER (A-Z)

0003 REAL FLAT, FLONG, NLAT, NLONG, H4, H5, H6, H7

0004 BYTE BUF(1), BUFFER(1), SCNID

0005 DIMENSION LADD(2), IPADD(4)

0006 COMMON /HIST/H4(256), H5(256), H6(256), H7(256)

0007 COMMON/COMH/ST, SE, LS, LE, NOCHAN, NOSAMP, SCNID(12),

1 SUNEL, SUNAZ, SNSHD, NREV, EXYR, EXDAY, DAY, HR, MIN,

2 SEC, SEQNO(2), FLAT, FLONG, NLAT, NLONG

0008 DATA IPADD/6, 6, 6, 7/

0009 DATA LADD/4, 5/

0010 LINE=0

0011 EOF=0

0012 IE=0

0013 ILINE=LINST

0014 LSS=1

0015 LSE=SE-ST+1

0016 INIT=0

0017 IF(ICCT.NE.1)GO TO 10

0018 IPIX1=IPIXST

0019 ISTRT=1

C

C

C

DO LOOP FOR READING LINES FROM TAPE

0020 10 DO 100 I=1,256

0021 DO 100 II=1,2

0022 LINE=LINE+1

0023 DO 80 ICHAN=1,4

0024 CALL LDCODE(LUN1, IFRM, BUFFER, BFSZ, ILINE, ICHAN, LSS, LSE,

1 BUF, EOF, PC, IE, INIT)

IF(EOF.EQ.1)GO TO 900

IF(IE.EQ.0)GO TO 20

WRITE(6,1200)ILINE, ICCT

FORMAT(1X, 'ERROR ON TAPE READ OF DATA FOR LINE', I5,

1 ' FOR CCT STRIP', I3)

GO TO 90

INDX=0

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:31:11	30-OCT-80	PAGE 2
SKPRD. FTN		/TR: BLOCKS/WR		
0031	.	IPIX=IPIX1		
	C			
	C	DO LOOP TO SELECT PIXELS FROM LINE		
	C			
0032		DO 65 J=1,35		
0033		DO 60 NPIX=ISTRT,4		
0034		INDX=INDX+1		
0035		BUF(INDX)=BUF(IPIX)		
0036		IPIX=IPIX+IPADD(NPIX)		
0037		IF(IPIX.GT.LSE)GO TO 70		
0038	60	CONTINUE		
0039		ISTRT=1		
0040	65	CONTINUE		
	C			
	C	MOVED ALL DATA TO BE USED FROM THIS LINE AND CHANNEL		
	C			
0041	70	CALL SKPWRT(LUN2,BUF,INDX,ICHAN,ICCT,LINE),		
0042	80	CONTINUE		
	C			
	C	INCREMENT LINE NUMBER		
	C			
0043	90	ILINE=ILINE+LADD(II)		
0044	100	CONTINUE		
	C			
	C	FINISHED ALL DATA FOR THIS STRIP		
	C			
0045		IPIX1=IPIX-LSE		
0046		ISTRT=NPIX+1		
0047		IF(ISTRT.GT.4)ISTRT=4		
0048		ILINE=3000		
0049		ICHAN=1		
0050		CALL LDCODE(LUN1,IFRM,BUFFER,BFSZ,ILINE,ICHAN,LSS,LSE,BUF, 1 EOF,PC,IE,INIT)		
0051		IF(EOF.NE.1)GO TO 910		
0052		RETURN		
	C			
	C	ERROR ENCOUNTERED EOF IN DATA SET		
	C			
0053	900	WRITE(6,1000)ICCT,ILINE		
0054	1000	FORMAT(1X,'ERROR: END OF FILE IN STRIP',I5,' AT LINE',I5)		
0055		IE=1		
0056		RETURN		
	C			
	C	ERROR DID NOT FIND END OF FILE AT END OF DATA		
	C			
0057	910	IE=1		
0058		WRITE(6,1100)ICCT		
0059	1100	FORMAT(1X,'ERROR: NO END OF FILE FOUND FOR STRIP',I5)		
12	0060	RETURN		
11	0061	END		
10				

ORIGINAL PAGE IS  
OF POOR QUALITY

PAGE 3

30-OCT-80

14:31:11

V02-31E

FORTRAN IV-PLUS

SKPRD. FIN

TR: BLOCKS/MR

PROGRAM SECTIONS

ATTRIBUTES

NUMBER NAME SITE

1 \$CODE1 001236 335 RW. I. CON. LCL  
3 \$IDATA 000324 106 RW. D. CON. LCL  
4 \$VARS 000052 21 RW. D. CON. LCL  
6 HIST 010000 2048 RW. D. DVR. BCL  
7 CONH 000100 32 RW. D. DVR. GBL

ENTRY POINTS

NAME TYPE ADDRESS NAME TYPE ADDRESS NAME TYPE ADDRESS

SKPRD 1-00.000

VARIABLES

NAME TYPE ADDRESS NAME TYPE ADDRESS NAME TYPE ADDRESS NAME TYPE ADDRESS  
BFSZ I\*2 F-000010\* DAY I\*2 7-000044 EDF I\*2 F-000012\* EXEAV I\*2 7-000042 EXVR I\*2 7-000040  
FLAT R\*4 7-000060 FLONG R\*4 7-000064 MORADR I\*2 F-000022\* MR I\*2 7-000046 I I\*2 4-000032  
ICT I\*2 F-000024\* ICHAN I\*2 4-000036 IE I\*2 F-000014\* IFRM I\*2 4-000020\* II I\*2 4-000034  
ILINE I\*2 4-000016 INDX I\*2 4-000042 INIT I\*2 4-000024 IPIX I\*2 4-000044 IPIHST I\*2 F-000030\*  
IPIX1 I\*2 4-000026 ISTRT I\*2 4-000030 J I\*2 4-000046 LE I\*2 7-000006 LINE I\*2 4-000014  
LINST I\*2 F-000026\* LS I\*2 7-000004 ISE I\*2 4-000022 LSS I\*2 4-000020 LUN1 I\*2 F-000002\*  
LUN2 I\*2 F-000004\* MIN I\*2 7-000050 NLAT R\*4 7-000070 NLONG R\*4 7-000074 NMOCHAN I\*2 7-000010  
MUSAMP I\*2 7-000012 NPIX I\*2 4-000050 NREV I\*2 7-000036 PC I\*2 4-000040 SE I\*2 7-000002  
SEC I\*2 7-000052 SNSHD I\*2 7-000034 SI I\*2 7-000000 SUNAZ I\*2 7-000032 SUMEL I\*2 7-000030

ARRAYS

NAME TYPE ADDRESS SIZE DIMENSIONS

BUF L\*1 F-000006\* 000001 0 (1)  
BUFFER L\*1 F-000016\* 000001 0 (1)  
H4 R\*4 6-000000 002000 512 (256)  
H5 R\*4 6-002000 002000 512 (256)  
H6 R\*4 6-004000 002000 512 (256)  
H7 R\*4 6-006000 002000 512 (256)  
I\*ADD I\*2 4-000004 000010 4 (4)  
LADD I\*2 4-000000 000004 2 (2)  
SCNID L\*1 7-000014 000014 6 (12)  
SENG I\*2 7-000054 000004 2 (2)

LABELS

NAME LABEL ADDRESS LABEL ADDRESS LABEL ADDRESS LABEL ADDRESS  
10 1-000154 20 1-000404 60 65 70 1-000566  
80 1-000632 100 900 1-001112 910 1-001170  
1000 3-000074 1100 3-000192 1200 3-000000

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTAN IV-PLUS V02-51E 14:31:11 30-OCT-80 PAGE 4  
SKPRD. FTN /TR: BLOCKS/WR

FUNCTIONS AND SUBROUTINES REFERENCED

LDCODE SKPWRT

TOTAL SPACE ALLOCATED = 011734 2542

NO FPP INSTRUCTIONS GENERATED

, TT17: [320, 40]A. B/-SP=SYO: [320, 40]SKPRD



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTTRAN IV-PLUS V02-51E 14:31:31 30-OCT-80 PAGE 1

SKPWRT.FTN /TR:BLOCKS/WR

0001 SUBROUTINE SKPWRT(LUN2,BUF,INDX,ICHAN,ICCT,LINE)

C

C

C

C

C

CALL ARGUMENTS:

C

C

C

C

C

C

C

C

0002 IMPLICIT INTEGER (A-Z)

0003 BYTE BUF(1),OBUF(2560),BWORD(2)

0004 EQUIVALENCE(BWORD,IWORD)

C

C

C

CHECK TO SEE IF FIRST CCT SU INITIALIZE ANCILLARY DATA

0005 IF(ICCT.EQ.4)INDX1=512-ICNT

0006 IF(ICCT.EQ.4.AND.INDX1.LT.INDX)INDX=INDX1

0007 IF(ICCT.NE.1)GO TO 10

0008 IF(ICHAN.NE.1)GO TO 20

0009 ICNT=0

0010 OBUF(2)=1

0011 IWORD=LINE

0012 OBUF(69)=BWORD(2)

0013 OBUF(70)=BWORD(1)

0014 GO TO 20

C

C

C

IF NOT CCT STRIP 1 READ DATA WRITTEN FOR STRIP 1

0015 IF(ICHAN.EQ.1)READ(LUN2'LINE)OBUF

C

C

C

DO LOOP FOR INSERTING DATA INTO PROPER PART OF OBUF

0016 DO 100 I=1,INDX

0017 IND=70+(ICHAN-1)\*512+ICNT+I

0018 OBUF(IND)=BUF(I)

0019 CONTINUE

C

C

C

IF CHANNEL IS 4 WRITE DATA BACK TO FILE

0020 IF(ICHAN.EQ.4)WRITE(LUN2'LINE)OBUF

C

C

C

IF LINE IS LAST LINE UPDATE ICNT

0021 IF(LINE.EQ.512.AND.ICHAN.EQ.4)ICNT=ICNT+INDX

C

C

C

IF LAST CCT AND CHAN 4 COMPUTE HISTOGRAM

0022 IF(ICHAN.EQ.4.AND.ICCT.EQ.4)CALL SKPHST(OBUF)

0023 RETURN

0024 END

FORTTRAN IV-PLUS V02-51E 14:31:31 30-OCT-80 PAGE 2  
SKPURT.FTN /TR:BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000502 161	RW,I,CON,LCL
3	\$1DATA	000026 11	RW,D,CON,LCL
4	\$VARS	005012 1285	RW,D,CON,LCL
5	\$TEMP'S	000002 1	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
SKPURT		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
1	I*2	4-005006	ICCT	I*2	F-000012*	ICHAN	I*2	F-000010*
INDX	I*2	F-000006*	INDX1	I*2	4-005002	WORD	I*2	4-000000
						LINE	I*2	F-000001*
						LUN2	I*2	F-000002*

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
BUF	L*1	F-000004*	000001	0 (1)
SKPURT	L*1	4-000000*	000002	1 (2)
ORUF	L*1	4-000002	005000	1280 (2560)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000170	20	1-000244	100	**

FUNCTIONS AND SUBROUTINES REFERENCED

SKPURT

TOTAL SPACE ALLOCATED = 005544 1458

NO FPP INSTRUCTIONS GENERATED

.IT17: [320.40]A. B/-SP=3V0: [320.40]SKPURT

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:31:52	30-OCT-80	PAGE 1
SKPHST.FTN		/TR:BLUCKS/WR		
0001		SUBROUTINE SKPHST(OBUF)		
	C			
	C	THIS ROUTINE WILL COMPUTE THE HISTOGRAM FOR THE SKIP IMAGE		
	C	PROCESSOR.		
	C			
	C	CALL ARGUMENTS:		
	C	OBUF	BUFFER CONTAINING PIXEL DATA TO COUNT	
	C			
0002		BYTE OBUF(1)		
0003		COMMON/HIST/(H4(256),H5(256),H6(256),H7(256))		
	C			
	C	ADD TO PROPER HISTOGRAM		
	C			
0004		DO 100 I=1,512		
0005		INDX1=OBUF(70+I).AND."377		
0006		INDX2=OBUF(582+I).AND."377		
0007		INDX3=OBUF(1094+I).AND."377		
0008		INDX4=OBUF(1606+I).AND."377		
0009		H4(INDX1+1)=H4(INDX1+1)+1		
0010		H5(INDX2+1)=H5(INDX2+1)+1		
0011		H6(INDX3+1)=H6(INDX3+1)+1		
0012		H7(INDX4+1)=H7(INDX4+1)+1		
0013	100	CONTINUE		
0014		RETURN		
0015		END		

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTAN IV-PLUS V02-31E 14:31:52 30-OCT-80 PAGE 2  
SKPHST.FTN /TR: BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	%CODE1	000260	B8 RW, I, CON, LCL
3	%DATA	000012	5 RW, D, CON, LCL
4	%VARS	000012	5 RW, D, CON, LCL
6	%HIST	010000	2048 RW, D, DVR, CBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
SKPHST		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
1	I*2	4-000000	INDX1	I*2	4-000002	INDX2	I*2	4-000004
						INDX3	I*2	4-000006
						INDX4	I*2	4-000010

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
H4	R*4	6-000000	002000	512 (256)
H5	R*4	6-002000	002000	512 (256)
H6	R*4	6-004000	002000	512 (256)
H7	R*4	6-006000	002000	512 (256)
COUF	L*1	F-000002*	000001	0 (1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
100	**				

TOTAL SPACE ALLOCATED = 010304 2146

.TT17 (320,40)A,B/SP-SY0:(320,40)SKPHST

## APPENDIX C

### LISTING OF READTP PROCESSOR





ORIGINAL PAGE IS  
OF POOR QUALITY

	FORTTRAN IV-PLUS V02-51E	14:20:21	30-OCT-80	PAGE 3
	READTP.FTN	/TR:BLOCKS/WR		
	C			
0041	IF(IE.NE.0)GOTO 40			
	C			
	C BUILD THE HEADER FILE NAME, AND THE IMAGE DATA FILE NAME			
	C			
0042	CALL INTOCT(EXDAY,OCTAL)			
0043	FLNIMH(12)=BWORD(2)			
0044	FLNIMH(13)=BWORD(1)			
0045	FLNIMD(12)=BWORD(2)			
0046	FLNIMD(13)=BWORD(1)			
0047	CALL INTOCT(VERSN,OCTAL)			
0048	FLNIMH(14)=BWORD(2)			
0049	FLNIMH(15)=BWORD(1)			
0050	FLNIMD(14)=BWORD(2)			
0051	FLNIMD(15)=BWORD(1)			
	C			
	C WRITE THE HEADER IN IMDACS FORMAT			
	C			
0052	CALL HDRFLN(LUN2,FLNIMH,UIFHR,SUNEL,FLAT,FLONG)			
	C			
	C OPEN THE IMAGE DATA FILE			
	C			
0053	OPEN(UNIT=LUN2,NAME=FLNIMD,TYPE='NEW',FORM='UNFORMATTED', + ACCESS='DIRECT',RECORDSIZE=640)			
	C			
	C FILL THE ENTIRE FILE WITH ZEROS			
	C			
0054	DO 16 M=1,512			
0055	WRITE(LUN2,M)PIXBUF			
0056	16 CONTINUE			
	C			
	C DETERMINE IF THE IMAGE TAPE IS DIFFERENT FROM THE HDT FULL FRAME			
	C			
0057	IF(INS.NE.3548)GOTO 30			
	C			
0058	20 CONTINUE			
0059	STRIPC=STRIPC+1			
	C			
	C THIS ROUTINE WILL BUILD A PATCH FRAME IMAGE USING HDT FULL FRAME C SIZE IMAGES			
	C			
0060	CALL HDT(LUN1,LUN2,BUF,BFSZ,NPOINT,FLNIMD,EOF,IE, + BUFFER,IFRM,ISS,INIT,STRIPC,PSIZE,INCL,INITL, + INCP,INITP)			
0061	IF(IE.NE.0)GOTO 40			
0062	IF(STRIPC.EQ.4)GOTO 50			
0063	IF(EOF.EQ.1)CALL HDRREAD(LUN1,IFRM,BUFFER,BFSZ,EOF, + PC,IE,BUFFER,INS,HDRADR,ISS)			
12 0064	IF(IE.NE.0)GOTO 40			
11 0065	ASSIGN 20 TO PROCES			
10 0066	IF(EOF.EQ.1)GOTO 35			
9 0067	INIT=0			
8 0068	GOTO 20			
7	C			
6 0069	30 CONTINUE			
5 0070	STRIPC=STRIPC+1			



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14 20:21 30-OCT-80 PAGE 4  
READTP.FTN /TR:BLOCKS/WR

C  
C THIS ROUTINE WILL BUILD A PATCH FRAME IMAGE USING LANDSAT  
C 1 & 2 FULL FRAME SIZE IMAGES  
C

0071 CALL LANSAT(LUN1,LUN2,BUF,BFSZ,NPOINT,FLNIMD,EOF,IE,BUFFER,  
+ IFRM,ISS,INIT,STRIPC,PSIZE,INCL,INITL,  
+ INCP,INITP)

0072 IF(IE.NE.0)GOTO 40

0073 IF(STRIPC.EQ.4)GOTO 50

0074 IF(EOF.EQ.1)CALL HDREAD(LUN1,IFRM,BUFFER,BFSZ,EOF,PC,IE,  
+ BUFFER,INS,HDRADR,ISS)

0075 IF(IE.NE.0)GOTO 40

0076 ASSIGN 30 TO PROCES

0077 IF(EOF.EQ.1)GOTO 35

0078 INIT=0

0079 GOTO 30

C  
C INFORM THE OPERATOR TO MOUNT THE NEXT TAPE IF IT EXISTS  
C

0080 35 CONTINUE

0081 CALL WTQIO("2540,LUN1,1,,ISTAT,AST)

0082 IF(ISTATB(1).LT.0)GOTO 40

0083 WRITE(4,6)BELL

0084 37 CONTINUE

C  
C CHECK TO SEE IF DATA TAPE IS READY  
C

0085 CALL WTQIO("2520,LUN1,1,,ISTAT,IDS)

0086 IF(ISTATB(4).NE."044")GOTO 37

0087 CALL WAIT(15000,1,M)

0088 INIT=0

0089 CALL HDREAD(LUN1,IFRM,BUFFER,BFSZ,EOF,PC,IE,  
+ BUFFER,INS,HDRADR,ISS)

0090 GOTO PROCES

0091 40 CONTINUE

0092 WRITE(6,4)IE,NAME

0093 GOTO 9999

0094 45 CONTINUE

0095 GOTO 9999

0096 50 CONTINUE

C  
C CLOSE THE IMAGE DATA FILE  
C

0097 CLOSE(UNIT=LUN2)

0098 CALL GIO("2400,LUN1)

C  
C WRITE THE HISTOGRAMS TO THE HEADER FILE  
C

0099 CALL HDRHIS(LUN2,FLNIMH,UIFHR)

0100 9999 STOP

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E

READTP. FTN

/TR: BLOCKS/WR

14:20:21

30-OCT-80

PAGE 5

0101

END

14:20:21 30-OCT-80 PAGE 6

FORTAN IV-PLUS V02-SIE  
READTP.FTN /TR: BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	CODE1	001402	385 RM, I. CON. LCL
2	SPDATA	000040	16 RM, D. CON. LCL
3	IDATA	000642	209 RM, D. CON. LCL
4	SVARS	030472	6301 RM, D. CON. LCL
6	COMH	000100	32 RM, D. DVR. CBL
7	LABEL	000012	5 RM, D. DVR. CBL
9	HIST	010000	2048 RM, D. DVR. CBL

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ANS	L*1	4-012450	AST	I*2	4-030466	BFSZ	I*2	4-030410	DAY	I*2	4-000044
DELTA	R*4	4-000012	E	I*2	7-000010	EUF	I*2	4-030450	EXDAY	I*2	4-000042
FLAT	R*4	6-000050	FLONG	R*4	6-000050	HDRADR	I*2	4-030444	HR	I*2	4-000046
IE	I*2	4-030454	IFRM	I*2	4-030446	INCL	I*2	4-030432	INCP	I*2	4-030436
INITL	I*2	4-030434	INITP	I*2	4-030440	INS	I*2	4-030456	ISS	I*2	4-030442
L	I*2	4-030422	LE	I*2	6-000066	LS	I*2	6-000004	LUN1	I*2	4-030416
M	I*2	4-030462	MIN	I*2	6-000050	NLAT	R*4	6-000070	NLONG	R*4	6-000074
NDSAMP	I*2	6-000012	NREV	I*2	6-000036	OCIAL	I*2	4-000004	PC	I*2	4-030452
PROCES	I*2	4-030464	PSIZE	I*2	4-030430	SE	I*2	6-000002	SEC	I*2	6-000052
ST	I*2	6-000000	STRIPC	I*2	4-030424	SUNAZ	I*2	6-000032	SUNEL	I*2	6-000030
MLB	I*2	4-030412							VERSN	I*2	4-030426

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
BELL	L*1	4-022357	000004	2 (4)
BUF	L*1	4-012517	007640	2000 (4000)
SUFFER	L*1	4-000022	006320	1640 (3280)
PARO	L*1	4-000004	000002	1 (2)
FILEH	L*1	4-012451	000006	3 (6)
FLNRD	L*1	4-012457	000020	8 (16)
FLNRH	L*1	4-012477	000020	8 (16)
H4	R*4	8-000000	002000	512 (256)
H5	R*4	8-002000	002000	512 (256)
H6	R*4	8-004000	002000	512 (256)
H7	R*4	8-006000	002000	512 (256)
IPRM	I*2	4-022370	000014	6 (6)
ISFAT	I*2	4-000000	000004	2 (2)
ISFATB	L*1	4-000000	000004	2 (4)
MISRO	L*1	7-000006	000002	1 (2)
NAME	I*2	7-000000	000006	3 (3)
NPOINT	I*2	4-022364	000004	2 (2)
OCT	I*2	4-022404	000004	2 (2)
PIABUF	L*1	4-006342	004106	105 (2118)
SENID	L*1	6-000014	000014	6 (12)
SENO	I*2	6-000054	000004	2 (2)
UIFIR	I*2	4-022410	006000	1536 (1536)

FORTRAN IV-PLUS V02-S1E 14:20:21 30-OCT-80 PAGE 7  
READTP.FIN /TR:BLOCKS/WR 1.

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1'	3-000000	4'	3-000056	5'	**	6'	3-000126
10	**	12	1-000344	15	**	16	**
30	1-001004	35	1-001116	37	1-001172	40	1-001250
50	1-001330	999	1-001366			45	1-001320

FUNCTIONS AND SUBROUTINES REFERENCED

CLOS\$ GETADR HDREAD HDRFLN HDRHIS HDT INTOCT LANSAT OPEN\$ QIO WAIT WTQIO

TOTAL SPACE ALLOCATED = 043110 8996

.TT17:[320,40]A.B/-SP=SYO:[320,40]READTP

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14:21:00    30-OCT-80      PAGE 1
LANSAT.FTN      /TR:BLOCKS/WR

0001      SUBROUTINE LANSAT(LUN1,LUN2,BUF,BFSZ,NPOINT,FLNIMD,EOF,IE,BUFFER,
+          IFRM,SS,INIT,STRIPC,PSIZE,INCL,INITL,INCP,
+          INITP)
C
C THIS ROUTINE WILL EXTRACT PATCH IMAGES FROM THE FULL FRAME IMAGE;
C PROCESSING TAKING PLACE ONE STRIP OF CCT AT A TIME. IT WILL
C FURTHER BUILD A PATCH FRAME IMAGE 16 PATCHES BY 16 PATCHES, EACH
C PATCH CONSISTING OF 32 BY 32 PIXELS, BANDS [4-7]. THE ROUTINE
C WILL USE LANDSAT 1 AND 2 FULL FRAME SIZE IMAGES.
C
C LOCAL VARIABLES:
C
C BLOCKN= THE BLOCK NUMBER
C ENDCCT= A PARAMETER FOR READING THE END OF THE FILE
C LCOUNT= THE LINE COUNTER FOR THE PATCH IMAGE
C LEND= THE ENDING LINE WITHIN THE FULL FRAME
C LSTART= THE STARTING LINE WITHIN THE FULL FRAME
C NPATCH= THE NUMBER OF PATCHES ON A SIDE OF A PATCH IMAGE
C PATCH= THE CURRENT PATCH NUMBER
C PIXELS= THE PIXEL LOCATION IN PIXBUF
C PTCH= A FLAG TO INDICATE THAT THE PATCH COUNTER HAS
C SAVEP= THE LAST PATCH PROCESSED ON PREVIOUS CCT
C STARTP= THE STARTING PIXEL LOCATION IN PATCH EXTRACTION
C STOPP= THE LAST PIXEL LOCATION IN PATCH EXTRACTION
C
C      BEEN READJUSTED
C
0002      IMPLICIT INTEGER (A-Z)
0003      LOGICAL NEXTL,NEXTP,PTCH
0004      REAL FLAT,FLONG,NLAT,NLONG,DELTAL,DELTAP,PO
0005      LOGICAL*1 BUF(1),PIXBUF(2118),FLNIMD(14),SCNID,MISNO(2)
0006      LOGICAL*1 BUFFER(1),BLOCK3(2)
0007      DIMENSION NPOINT(2),ISTAT(2)
0008      EQUIVALENCE (BLOCKB,BLOCKN)
0009      COMMON/COMH/ST,SE,LS,LE,NOCHAN,NOSAMP,SCNID(12),
+          SUNEL,SUNAZ,SNSHD,NREV,EXYR,EXDAY,DAY,
+          HR,MIN,SEC,SENO(2),FLAT,FLONG,NLAT,NLONG
0010      COMMON/LABEL/NAME(3),MISNO,E
0011      DATA SAVEP/O/,TSE/O/,TST/O/
C
C      INITIALIZING PARAMETERS
C
0012      LSS=1
0013      IF(IFRM.NE.3)TSE=SE
0014      IF(IFRM.EQ.3)TST=TSE+1
0015      IF(IFRM.EQ.3)TSE=TSE+SE
0016      IF(IFRM.NE.3)TST=ST
0017      LSE=SE-ST+1
0018      EOF=0
0019      DO 5 M=71,2118
0020      PIXBUF(M)=0
0021      5 CONTINUE
0022      NPATCH=512/PSIZE
C
C      CHECKING FOR THE CASE WHERE ALL THE PATCHES HAVE BEEN
C      EXTRACTED
C

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:21:00	30-OCT-80	PAGE 2
LANSAT.FTN		/TR: BLOCKS/WR		
0023	IF(SAVEP.EQ.NPATCH)GOTO 9999			
0024	PCOUNT=SAVEP			
	C			
	C SET START AND STOP PIXEL FOR READING EACH STRIP FROM CCT			
	C SET START LINE FOR EXTRACTION FROM CCT			
	C			
	C PO=-1*(.255+.00303*FLAT)			
	C DELTA=(NPOINT(1)-FLAT)*23.073			
	C DELTAP=(NPOINT(2)-FLONG)*COS(FLAT)*32.99			
	C NOMP=1519.+DELTAP+DELTAL*PO			
	C NOML=1170.+DELTAL			
0025	WRITE(6,1111)DELTAL,DELTAP			
0026	1111 FORMAT(1X,'DELTAL=',1F10.5,'DELTAP=',1F10.5)			
	C			
0027	DO 90 M=1,NPATCH			
0028	LCOUNT=0			
0029	LINE=M*PSIZE			
0030	STARTL=INITL+INCL*(M-1)			
	C			
	C THIS ROUTINE WILL CHECK FOR THE CASE WHERE THE STARTING			
	C AND ENDING LINE OF PATCHES ARE OUT OF THE FULL FRAME			
	C			
0031	CALL LBORDR(STARTL,NEXTL,LS,LE,LEND,LSTART,PSIZE)			
0032	IF(NEXTL)GOTO 80			
	C			
	C PROCESS LINES FOR EACH PATCH			
	C			
0033	DO 70 K=LSTART,LEND			
0034	LCOUNT=LCOUNT+1			
0035	BLOCKN=LINE-PSIZE+(LSTART-1)+LCOUNT			
	C			
	C PROCESSING ONE BAND AT A TIME FOR EACH LINE IN PATCH			
	C			
0036	DO 60 I=1,4			
0037	SAVEP=PATCH-1			
0038	PATCH=PCOUNT			
0039	RCHAN=I			
0040	PTCH=.TRUE.			
	C			
	C THIS ROUTINE WILL READ ONE LINE OF IMAGE DATA ONE			
	C CHANNEL AT A TIME			
	C			
0041	CALL LDCODE(LUN1,IFRM,BUFFER,BFS2,STARTL,RCHAN,LSS,			
	+ LSE,BUF,EOF,PC,IE,INIT)			
	C			
	C CHECKING FOR READ ERROR			
	C			
0042	IF(EOF.EQ.1)IE=1			
0043	IF(IE.NE.0)GOTO 9999			
12				
11	C			
10	0044 10 CONTINUE			
9	0045 PATCH=PATCH+1			
8	C			
7	C			
6	C CALCULATING THE NUMBER OF PIXELS PROCESSED PER PATCH			
5	C CALCULATING START AND STOP PIXELS PER PATCH			
4	C			

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:21:00	30-OCT-80	PAGE 3
LANSAT.FTN		/TR.BLOCKS/WR		
0046		PIXELS=(PATCH-1)*PSIZE+(70+512*(I-1))		
0047		STARTP=INITP+INCP*(PATCH-1)		
0048		IF(STARTP.GT.TSE.OR.PATCH.GT.NPATCH)GOTO 60		
0049		STOPP=STARTP+PSIZE-1		
	C			
	C	THIS ROUTINE WILL CHECK THE BOUNDARY OF THE PATCH		
	C	TO DETERMINE IF IT'S IN THE FULL FRAME		
	C			
0050		CALL PBORDR(STARTP,STOPP,STOP,NEXTP,TSE,TST,PSIZE)		
0051		IF(NEXTP)GOTO 10		
	C			
	C	THIS ROUTINE WILL CHECK FOR THE CASE WHERE A PATCH		
	C	IS DIVIDED BY TWO CCT'S		
	C			
0052		CALL PARTP(PATCH,STARTP,STOPP,STOP,STRIPC,TSE,TST,		
		PIXELS,I,PATCH,PSIZE)		
	C			
	C	READ PIXEL ARRAY FROM .IMD FILE		
	C			
0053		READ(2'BLOCKN)PI/BUF		
	C			
	C	PLACE THE DESIRED PIXELS IN THE PIXEL ARRAY		
	C			
0054		L=0		
0055		DO 50 J=STARTP,STOPP		
0056		L=L+1		
0057		N=STARTP-TST+L		
0058		PIXBUF(PIXELS+L)=BUF(N)		
0059	50	CONTINUE		
	C			
	C	PROCESS DATA EXTRACTED FOR HISTOGRAM		
	C			
0060		CALL HIST(BUF,STARTP,STOPP,RCHAN,TST)		
	C			
	C	WRITE THE PIXEL ARRAY BACK TO .IMD FILE		
	C			
0061		PIXBUF(2)=1		
0062		PIXBUF(69)=BLOCKB(2)		
0063		PIXBUF(70)=BLOCKB(1)		
0064		WRITE(2'BLOCKN)PIXBUF		
	C			
0065		GOTO 10		
	C			
0066	60	CONTINUE		
	C			
0067	70	STARTL=STARTL+1		
0068		GOTO 90		
	C			
0069	80	CONTINUE		
	C			
	C	WRITING ZERO FILLED PATCH LINES		
	C			
0070		DO 85 J=71,2118		
0071		PIXBUF(J)=0		
0072	85	CONTINUE		
	C			

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:21:00	30-OCT-80	PAGE 4
LANSAT. PTN /TR: BLOCKS/WR				
0073	DO 87 J=1,PSIZE			
0074	BLOCKN=BLOCKN+1			
0075	PIXBUF(2)=1			
0076	PIXBUF(69)=BLOCKB(2)			
0077	PIXBUF(70)=BLOCKB(1)			
0078	WRITE(LUN2,BLOCKN)PIXBUF			
0079	87 CONTINUE			
	C			
0080	90 CONTINUE			
	C			
	C READ UNTIL END OF CCT STRIP			
	C			
0081	ENDCCT=3000			
0082	CALL LDCODE(LUN1,IFRM,BUFFER,BFSZ,ENDCCT,RCHAN,LSS,LSE, + BUF,EOF,PC,IE,INIT)			
	C			
0083	9999 RETURN			
0084	END			



PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1 002004	514	RM, I, CON, LCL
3	\$IDATA 000306	99	RM, D, CON, LCL
4	\$VARS 004222	1097	RM, D, CON, LCL
5	\$TEMP5 000006	3	RM, D, CON, LCL
6	COMH 000100	32	RM, D, DVR, GBL
7	LABEL 000012	5	RM, D, DVR, GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LANSAT	1	000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BFSZ	I*2	F-000010*	BLOCKN	I*2	4-000000	DAY	I*2	6-000044
E	I*2	7-000010	ENDCCY	I*2	4-004220	EDF	I*2	F-000016*
FLAT	R*4	6-000050	FLDNG	R*4	6-000054	HR	I*2	6-000046
IFRM	I*2	F-000024*	INCL	I*2	F-000036*	INCP	I*2	F-000042*
INITP	I*2	F-000004*	J	I*2	4-004214	K	I*2	4-004170
LE	I*2	6-000006	LEND	I*2	4-004164	LINE	I*2	4-004160
LSS	I*2	4-004144	LSTART	I*2	4-004166	LUNI	I*2	F-000002*
NIN	I*2	6-000050	N	I*2	4-004216	NEXTL	I*2	4-000002*
NLNG	R*4	6-000074	NDCHAN	I*2	6-000010	NOSAMP	I*2	6-000012
PATCH	I*2	4-004174	PC	I*2	4-004200	PCOUNT	I*2	4-004154
PSIZE	I*2	F-000034*	PICH	I*2	4-000006	RCHAN	I*2	4-004176
SEC	I*2	6-000052	SNSHD	I*2	6-000034	SS	I*2	F-000026*
STARTP	I*2	4-004304	STOP	I*2	4-004210	STOPP	I*2	4-004206
SUNEL	I*2	6-000030	TSE	I*2	4-004140	TST	I*2	4-004142

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
BLOCKB	L*1	4-000000	000002	1 (2)
BUF	L*1	F-000004*	000001	0 (1)
BUFFER	L*1	F-000022*	000001	0 (1)
FLINHD	L*1	F-000014*	000016	7 (14)
ISAT	I*2	4-004132	000004	2 (2)
MISNO	L*1	7-000006	000002	1 (2)
NAME	I*2	7-000000	000006	3 (3)
NPPOINT	I*2	F-000012*	000004	2 (2)
PIRDR	L*1	4-000024	004106	1059 (2118)
SCNID	L*1	6-000014	000014	6 (12)
SEGRD	I*2	6-000054	000004	2 (2)

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY

FOR: RARI IV-PLUS V02-51E 14:21:00 30-OCT-80. PAGE 6  
L:R:JAT FTR 7TR:BLOCKS/MR

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
5	**	10	1-000766	50	**	60	1-001426
80	1-001502	85	**	87	**	90	1-001632
999	1-002002					1111	3-000000

FUNCTIONS AND SUBROUTINES REFERENCED

HIST LBORDR LDCODE PARTP PBORDR

TOTAL SPACE ALLOCATED = 006654 1750

NO FPP INSTRUCTIONS GENERATED

.1117 (320.40)A B/-SP=SYO (320.40)ANSAT

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14 21 58 30-OCT-80 PAGE 1  
HDT. FTN /TR. BLOCKS/WR

0001 SUBROUTINE HDT(LUN1, LUN2, BUF, BFSZ, FLNIMD, EOF, IE, BUFFER, IFRM,  
+ SS, INIT, STRIPC, PSIZE, INCL, INIL, INCP, INITP)

C  
C THIS ROUTINE WILL EXTRACT PATCH IMAGES FROM THE FULL FRAME IMAGE,  
C PROCESSING TAKING PLACE ONE STRIP OF CCT AT A TIME. IT WILL  
C FURTHER BUILD A PATCH FRAME IMAGE 16 PATCHES BY 16 PATCHES, EACH  
C PATCH CONSISTING OF 32 BY 32 PIXELS, BANDS [4-7]. THE ROUTINE  
C WILL USE HDT FULL FRAME SIZE IMAGES.

C LOCAL VARIABLES:

C  
C BLOCKN= THE BLOCK NUMBER  
C ENDCCT= A PARAMETER FOR READING THE END OF THE FILE  
C LCOUNT= THE LINE COUNTER FOR THE PATCH IMAGE  
C LEND= THE ENDING LINE WITHIN THE FULL FRAME  
C LSTART= THE STARTING LINE WITHIN THE FULL FRAME  
C NPATCH= THE NUMBER OF PATCHES ON A SIDE OF A PATCH IMAGE  
C PATCH= THE CURRENT PATCH NUMBER  
C PIXELS= THE PIXEL LOCATION IN PIXBUF  
C PTCH= A FLAG TO INDICATE THAT THE PATCH COUNTER HAS  
C SAVEP= THE LAST PATCH PROCESSED ON PREVIOUS CCT  
C STARTP= THE STARTING PIXEL LOCATION IN PATCH EXTRACTION  
C STOPP= THE LAST PIXEL LOCATION IN PATCH EXTRACTION  
C BEEN READJUSTED

0002 IMPLICIT INTEGER (A-Z)

0003 LOGICAL NEXTL, NEXTP, PTCH

0004 REAL FLAT, FLONG, NLAT, NLONG

0005 LOGICAL\*1 B/F(1), PIXBUF(2118), FLNIMD(14), SCNID, MISNO(2)

0006 LOGICAL\*1 BUFFER(1), BLOCKB(2)

0007 DIMENSION ISTAT(2)

0008 EQUIVALENCE (BLOCKB, BLOCKN)

0009 COMMON/COMH/ST, SE, LS, LE, NOCHAN, NOSAMP, SCNID(12),  
+ SUNEL, SUNAZ, SNSHD, NREV, EXYR, EXDAY, DAY,

+ HR, MIN, SEC, SEGNO(2), FLAT, FLONG, NLAT, NLONG

0010 COMMON/LABEL/NAME(3), MISNO, E

0011 DATA SAVEP/0/, TSE/0/, TST/0/

C  
C INITIALIZING PARAMETERS

C

0012 LSS=1

0013 IF(IFRM.NE.3)TSE=SE

0014 IF(IFRM.EQ.3)TST=TSE+1

0015 IF(IFRM.EQ.3)TSE=TSE+SE

0016 IF(IFRM.NE.3)TST=ST

0017 LSE=SE-ST+1

0018 EOF=0

0019 DO 5 M=71, 2118

0020 PIXBUF(M)=0

0021 5 CONTINUE

0022 NPATCH=512/PSIZE

0023 PCOUNT=SAVEP

C

C SET START AND STOP PIXEL FOR READING EACH STRIP FROM CCT

C SET START LINE FOR EXTRACTION FROM CCT

C

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14 21 58	30-OCT-80	PAGE 2
HDT FTH		/TR BLOCKS/WR		
0024	DO 90 M=1,NPATCH			
0025	LCOUNT=0			
0026	LINE=M*PSIZE			
0027	STARTL=INITL+INCL*(M-1)			
	C			
	C	THIS ROUTINE WILL CHECK FOR THE CASE WHERE THE STARTING		
	C	AND ENDING LINE OF PATCHES ARE OUT OF THE FULL FRAME		
	C			
0028	CALL LBORDR(STARTL,NEXTL,LS,LE,LEND,LSTART,PSIZE)			
0029	IF(NEXTL)GOTO 80			
	C			
	C	PROCESS LINES FOR EACH PATCH		
	C			
0030	DO 70 K=LSTART,LEND			
0031	LCOUNT=LCOUNT+1			
0032	BLOCKN=LINE-PSIZE+(LSTART-1)+LCOUNT			
	C			
	C	PROCESSING ONE BAND AT A TIME FOR EACH LINE IN PATCH		
	C			
0033	DO 60 I=1,4			
0034	SAVEP=PATCH-1			
0035	PATCH=PCOUNT			
0036	RCHAN=1			
0037	PTCH=.TRUE.			
	C			
	C	THIS ROUTINE WILL READ ONE LINE OF IMAGE DATA ONE		
	C	CHANNEL AT A TIME		
	C			
0038	CALL LDCODE(LUN1,IFRM,BUFFER,BFSZ,STARTL,RCHAN,LSS, LSE,BUF,EOF,PC,IE,INIT)			
	C			
	C	CHECKING FOR READ ERROR		
	C			
0037	IF(EOF.EQ.1)IE=1			
0040	IF(IE.NE.0)GOTO 9999			
	C			
0041	10 CONTINUE			
0042	PATCH=PATCH+1			
	C			
	C	CALCULATING THE NUMBER OF PIXELS PROCESSED PER PATCH		
	C	CALCULATING START AND STOP PIXELS PER PATCH		
	C			
0043	PIXELS=(PATCH-1)*PSIZE+(70*512*(I-1))			
0044	STARTP=INITP+INCP*(PATCH-1)			
0045	IF(STARTP.GT.TSE)GOTO 60			
0046	STOPP=STARTP+PSIZE-1			
	C			
	C	THIS ROUTINE WILL CHECK THE BOUNDARY OF THE PATCH		
	C	TO DETERMINE IF IT'S IN THE FULL FRAME		
	C			
0047	CALL PBORDR(STARTP,STOPP,STOP,NEXTP,TSE,TST,PSIZE)			
0048	IF(NEXTP)GOTO 10			
	C			
	C	THIS ROUTINE WILL CHECK FOR THE CASE WHERE A PATCH		
	C	IS DIVIDED BY TWO CCT'S		
	C			

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14.21.58	30-OCT-80	PAGE 3
HDT.FTN		/TR:BLOCKS/WR		
0049		CALL PARTP(PATCH, STARTP, STOPP, STOP, STRIPC, TSE, TST, PIXELS, I, PATCH, PSIZE)		
	C			
	C	READ PIXEL ARRAY FROM .IMD FILE		
	C			
0050		READ(2'BLOCKN)PIXBUF		
	C			
	C	PLACE THE DESIRED PIXELS IN THE PIXEL ARRAY		
	C			
0051		L=0		
0052		DO 50 J=STARTP, STOPP		
0053		L=L+1		
0054		N=STARTP-TST+L		
0055		PIXBUF(PIXELS+L)=BUF(N)		
0056	50	CONTINUE		
	C			
	C	PROCESS DATA EXTRACTED FOR HISTOGRAM		
	C			
0057		CALL HIST(BUF, STARTP, STOPP, RCHAN, TST)		
	C			
	C	WRITE THE PIXEL ARRAY BACK TO .IMD FILE		
	C			
0058		PIXBUF(2)=1		
0059		PIXBUF(69)=BLOCKB(2)		
0060		PIXBUF(70)=BLOCKB(1)		
0061		WRITE(2'BLOCKN)PIXBUF		
	C			
0062		GOTO 10		
	C			
0063	60	CONTINUE		
	C			
0064	70	STARTL=STARTL+1		
0065		GOTO 90		
	C			
0066	80	CONTINUE		
	C			
	C	WRITING ZERO FILLED PATCH LINES		
	C			
0067		DO 85 J=71, 2118		
0068		PIXBUF(J)=0		
0069	85	CONTINUE		
	C			
0070		DO 87 J=1, PSIZE		
0071		BLOCKN=BLOCKN+1		
0072		PIXBUF(2)=1		
0073		PIXBUF(69)=BLOCKB(2)		
0074		PIXBUF(70)=BLOCKB(1)		
0075		WRITE(LUN2'BLOCKN)PIXBUF		
12 0076	87	CONTINUE		
11	C			
10 0077	90	CONTINUE		
9	C			
8	C	READ UNTIL END OF CCT STRIP		
7	C			
6 0078		ENDCCT=3000		
5 0079		CALL LDCODE(LUN1, IFRM, BUFFER, BFSZ, ENDCCT, RCHAN, LSS, LSE,		

ORIGINAL PAGE  
OF POOR QUALITY

FCTRAN IV-PLUS V02-51E

14-21-58

30-OCT-80

PAGE 4

HDT FTH

/TR: BLOCKS/WR

BUF, EOP, PC, YE, INIT)

C

0080

9999 RETURN

0081

END

-----HOT FTN-----/TR BLOCKS/WA

## PROGRAM SECTIONS

RUNNER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	475	RM. I. CON. LCL
3	\$IDATA	81	RM. D. CON. LCL
4	\$VARS	1091	RM. D. CON. LCL
5	\$TEMPS	3	RM. D. CON. LCL
6	CORH	32	RM. D. OVR. GBL
7	LABEL	5	RM. D. OVR. GBL

### ENTRY POINTS

[illegible]

## VARIABLES

[illegible]

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
BLCKB	L*1	4-000000	000002	1 (2)
BUF	L*1	F-000006*	000001	0 (1)
BU-FER	L*1	F-000020*	000001	0 (1)
FL71MD	L*1	F-000012*	000016	7 (14)
LEPAY	I*2	4-004116	000004	2 (2)
HL*20	L*1	7-000006	000002	1 (2)
JANE	I*2	7-000000	000006	3 (3)
PIFBUF	L*1	4-000010	004106	1059 (2118)
SCRIB	L*1	6-000014	000014	6 (12)
SEMD	I*2	6-000054	000004	2 (2)

LABELS

**ORIGINAL PAGE IS  
OF POOR QUALITY**

C-19

FORTAN IV-PLUS V02-51E		14:21:58		30-OCT-80		PAGE 6	
HDT.FTN		/TR:BLOCKS/LR					
LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
5	**	10	1-000662	50	**	60	1-001310
85	1-001364	85	**	87	**	90	1-001534
						9999	1-001664
FUNCTIONS AND SUBROUTINES REFERENCED							
HIST	LBORDR	LDCODE	PARTP	PBORDR			
TOTAL SPACE ALLOCATED " 006456 1687							
NO FPP INSTRUCTIONS GENERATED							
.IT17:[320,40]A B/-SP=SYO:[320,40]HDT							



ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E' 14:22:19 30-OCT-80 PAGE 1
PBORDR.FTN /TR.BLOCKS/WR

0001 SUBROUTINE PBORDR(STARTP,STOPP,STOP,NEXTP,TSE,TST,PSIZE)
      C
      C THIS ROUTINE WILL DETERMINE WHEN IMAGERY DATA DESIRED FROM
      C THE FULL FRAME IN NOT AVAILABLE; PATCH PIXELS ARE OUTSIDE
      C OF THE FULL FRAME
      C
      C LOCAL VARIABLES:
      C
      C NEXTP= A FLAG WHICH INDICATES; PROCESS THE NEXT PATCH
      C
0002 IMPLICIT INTEGER (A-Z)
0003 LOGICAL NEXTP
      C
      C CHECKING FOR THE CASE WHERE PATCH PIXELS ARE NOT WITHIN
      C THE FULL FRAME
      C
0004 NEXTP=.FALSE.
0005 IF(STARTP.GE.TST.AND.STARTP.LE.(TSE-PSIZE-1))GOTO 9999
0006 IF(STARTP.GT.TST)GOTO 40
0007 IF((STARTP+PSIZE-1).GT.TST)GOTO 10
0008 NEXTP=.TRUE.
0009 GOTO 9999
      C
0010 10 CONTINUE
      C
      C CALCULATE THE NUMBER OF PATCH PIXELS WHICH ARE OUT OF
      C THE LEFT BORDER OF THE FULL FRAME
      C
0011 DO 20 K=1,PSIZE
0012 IF((STARTP+K-1).EQ.TST)GOTO 30
0013 20 CONTINUE
      C
0014 30 CONTINUE
0015 STARTP=TST
0016 STOPP=STARTP+PSIZE-K
0017 NEXTP=.FALSE.
0018 GOTO 9999
      C
0019 40 CONTINUE
      C
      C CALCULATE THE NUMBER OF PATCH PIXELS WHICH ARE OUT OF
      C THE RIGHT BORDER OF THE FULL FRAME
      C
0020 IF(STARTP.GT.TSE)GOTO 60
0021 DO 50 K=1,PSIZE
0022 IF((STARTP+K-1).EQ.TSE)GOTO 70
0023 50 CONTINUE
      C
0024 60 CONTINUE
0025 NEXTP=.TRUE.
0026 GOTO 9999
      C
0027 70 CONTINUE
0028 STOP=STOPP
0029 STOPP=K+STARTP-1
0030 NEXTP=.FALSE.

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E

14:22:19

30-OCT-80

PAGE 2

PBORDR.FTN

/TR:BLOCKS/WR

0031

<sup>C</sup>9999 RETURN

0032

END

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:22:19 30-OCT-80 PAGE 3  
PBORDR.FTN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1 000404	130	RM.I, CON, LCL
4	\$VARS 000002	1	RM.D, CON, LCL
5	\$TEMPS 000002	1	RM.D, CON, LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
PBORDR		1-000000									

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
K	I*2	4-000000	NEXTP	I*2	F-000010*	PSIZE	I*2	F-000016*	STARTP	I*2	F-000002*
STOPP	I*2	F-000004*	TSE	I*2	F-000012*	TST	I*2	F-000014*			

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000116	20	**	30	1-000176	40	1-000240	50	**
50	1-000330	70	1-000344	9999	1-000402				

TOTAL SPACE ALLOCATED = 000410 132

NO FPP INSTRUCTIONS GENERATED

.TT17: (320, 40)A, B/-SP=SY0: (320, 40)PBORDR

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E  
LBORDR. FTN /TR: BLOCKS/WR

14:22:37 30-OCT-80

PAGE 1

```

0001      SUBROUTINE LBORDR(STARTL,NEXTL,LS,LE,LEND,LSTART,PSIZE)
      C
      C THIS ROUTINE WILL DETERMINE WHEN IMAGERY DATA DESIRED FROM FULL FRAME
      C IS NOT AVAILABLE; PATCH LINES ARE OUTSIDE OF THE FULL FRAME.
      C
      C LOCAL VARIABLES:
      C
      C LEND= LINE END
      C LSTART= LINE START
      C NEXTL= A FLAG WHICH INDICATES, PROCESS THE NEXT FULL FRAME LINE
      C
0002      IMPLICIT INTEGER (A-Z)
0003      LOGICAL NEXTL
      C
      C INITIALIZING PARAMETERS
      C
0004      LEND=PSIZE
0005      LSTART=1
      C
      C CHECKING FOR THE CASE WHERE PATCH LINES ARE NOT WITHIN THE FULL FRAME
      C
0006      NEXTL=.FALSE.
0007      IF(STARTL.GE.LS.AND.STARTL.LE.(LE-PSIZE-1))GOTO 9999
0008      IF(STARTL.GT.LS)GOTO 50
0009      IF((STARTL+PSIZE-1).GT.LS)GOTO 20
0010      NEXTL=.TRUE.
0011      GOTO 9999
      C
0012      20 CONTINUE
      C
      C CALCULATE THE NUMBER OF PATCH LINES WHICH ARE OUT OF THE
      C TOP BORDER OF THE FULL FRAME
      C
0013      DO 30 K=1,PSIZE
0014      IF((STARTL+K-1).EQ.LS)GOTO 40
0015      30 CONTINUE
      C
0016      40 CONTINUE
0017      LSTART=K
0018      NEXTL=.FALSE.
0019      GOTO 9999
      C
0020      50 CONTINUE
      C
      C CALCULATE THE NUMBER OF PATCH LINES WHICH ARE OUT OF THE
      C BOTTOM BORDER OF THE FULL FRAME
      C
0021      IF(STARTL.GT.LE)GOTO 70
0022      DO 60 K=1,PSIZE
0023      IF((STARTL+K-1).EQ.LE)GOTO 80
0024      60 CONTINUE
      C
0025      70 CONTINUE
0026      NEXTL=.TRUE.
0027      GOTO 9999
      C

```

ORIGINAL PAGE 1  
OF POOR QUALITY

FORTAN IV-PLUS V02-51E  
LBORDR. FTN /TR: BLOCKS/WR

14:22:37 30-OCT-80

PAGE 2

0028 80 CONTINUE  
0029 LEND=K+LSTART-1  
0030 NEXTL= FALSE.

C  
0031 9999 RETURN  
0032 END

ORIGINAL PAGE 13  
OF POOR QUALITY

FORTRAN IV-PLUS V02-SIE					14:22:37		30-OCT-80		PAGE 3		
LBORDR.FIN					/TR:BLOCKS/MR						
PROGRAM SECTIONS											
NUMBER		NAME		SIZE		ATTRIBUTES					
1		\$CODE1		000372		125		RM, I, CON, LCL			
4		\$VARS		000002		1		RM, D, CON, LCL			
5		\$STEPS		000002		1		RM, D, CON, LCL			
ENTRY POINTS											
NAME		TYPE		ADDRESS		NAME		TYPE		ADDRESS	
LBORDR				1-000000							
VARIABLES											
NAME		TYPE		ADDRESS		NAME		TYPE		ADDRESS	
K		I*2		4-000000		LE		I*2		F-000010* LEND I*2 F-000012* LS I*2 F-000006* LSTART I*2 F-000014*	
NEXTL		I*2		F-000004*		PSIZE		I*2		F-000016* STARTL I*2 F-000002*	
LABELS											
LABEL		ADDRESS		LABEL		ADDRESS		LABEL		ADDRESS	
20		1-000132		30				40		1-000212	
70		1-000324		80		1-000340		9999		1-000370	
										60	
										**	
TOTAL SPACE ALLOCATED = 000376 127											
NO F/P INSTRUCTIONS GENERATED											
.TT17:(320,40)A B/-SP=SY0:(320,40)LBORDR											

ORIGINAL PAGE 13  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14: 25: 36      30-OCT-80      PAGE 1
HDRFLN.FTN      /TR: BLOCKS/WR

0001      C      SUBROUTINE HDRFLN(LUN2,FLNIMH,UIFHR,SUNEL,FLAT,FLONG)
C      THIS ROUTINE WILL WRITE A HEADER IN IMDCS FORMAT CONSISTING OF A FILE
C      HEADER, A UNIVERSAL IMAGERY HEADER(SPU) FORMAT HEADER RECORD. EACH
C      IS 1536 WORDS LONG.
C
0002      IMPLICIT INTEGER (A-Z)
0003      REAL FLAT,FLONG
0004      LOGICAL*1 FLNIMH(16)
0005      DIMENSION UIFHR(1536)
C
C      OPEN THE HEADER FILE AND WRITE FIRST RECORD
C
0006      OPEN(UNIT=LUN2,NAME=FLNIMH,TYPE='NEW',FORM='UNFORMATTED',
+      ACCESS='DIRECT',RECORDSIZE=768)
0007      WRITE(LUN2,1)UIFHR
C
C      CLEAR THE HEADER BUFFER
C
0008      DO 10 K=1,1536
0009      UIFHR(K)=0
0010      10 CONTINUE
C
C      THIS ROUTINE WILL WRITE THE UNIVERSAL (SPU) PORTION OF THE HEADER
C
0011      CALL UNVRSL(LUN2,SUNEL,FLAT,FLONG)
C
C      CLOSE THE HEADER FILE
C
0012      CLOSE(UNIT=LUN2)
C
0013      RETURN
0014      END

```

FORTRAN IV-PLUS V02-51E 14:25:36 30-OCT-80 PAGE 2  
HURFLN F IN /TR BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000272	93 RM. I. CON. LCL
3	\$IDATA	000076	31 RM. D. CON. LCL
4	\$VARS	000002	1 RM. D. CON. LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
HURFLN		1-000000									

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
FLAT	R*4	F-000012*	FLONG	R*4	F-000014*	K	1*2	4-000000	LUN2	1*2	F-000002*
									SUNEL	1*2	F-000010*

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
FLNIMH	L*1	F-000004*	000020	B (16)
UIFMR	1*2	F-000006*	006000	1536 (1536)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	**						

FUNCTIONS AND SUBROUTINES REFERENCED

CLOS\* OPEN\* UNVRSI

TOTAL SPACE ALLOCATED = 000372 125

NO FPP INSTRUCTIONS GENERATED

IT117 (320.40)A. B/-SP-SY0 (320.40)HURFLN



ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-91E      14 25 93      30-OCT-90      PAGE 1
HDRHIS FTY      7TR BLOCKS/WR

0001      SUBROUTINE HDRHIS(LUN2,FLNIMH,UIFHR)
C
C THIS ROUTINE WILL WRITE THE HISTOGRAMS FOR EACH BAND (4-7)
C TO THE IMPACS HEADER FILE
C
0002      IMPLICIT INTEGER (A-Z)
0003      REAL H4,H5,H6,H7
0004      LOGICAL*1 FLNIMH(1)
0005      DIMENSION UIFHR(1536)
0006      COMMON/HIST/H4(256),H5(256),H6(256),H7(256)
C
C OPEN THE IMPACS HEADER FILE
C
0007      OPEN(UNIT=LUN2,NAME=FLNIMH,TYPE='OLD',FORM='UNFORMATTED',
+ ACCESS='DIRECT',RECORDSIZE=768)
C
C WRITE THE HISTOGRAMS FOR BANDS (4-7)
C
0008      WRITE(LUN2'3')H4,H5,H6
0009      WRITE(LUN2'4')H7
C
C THE REMAINING BAND HISTOGRAMS ARE NOT USED IN THIS APPLICATION
C
0010      DO 10 I=5,10
0011      WRITE(LUN2'I')UIFHR
0012      10 CONTINUE
C
C CLOSE THE IMPACS HEADER FILE
C
0013      CLOSE(UNIT=LUN2)
C
0014      RETURN
0015      END

```

FORTRAN IV-PLUS V02-51E 14:25:53 30-OCT-80 PAGE 2  
HDRHIS.FIN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	%CODE1	000320 104	RM, I, CON, LCL
3	%DATA	000124 42	RM, D, CON, LCL
4	%VARS	000002 1	RM, D, CON, LCL
6	HIST	010000 2048	RM, D, OVR, CBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
HDRHIS		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
I	I*2	4-000000	LUN2	I*2	F-000002*			

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
FLNIMH	L*1	F-000004*	000001	0 (1)
H4	R*4	6-000000	002000	512 (256)
H5	R*4	6-002000	002000	512 (256)
H6	R*4	6-004000	002000	512 (256)
H7	R*4	6-006000	002000	512 (256)
U1FMR	I*2	F-000006*	006000	1536 (1536)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	**						

FUNCTIONS AND SUBROUTINES REFERENCED

CLOS\* OPENS

12 TOTAL SPACE ALLOCATED = 010446 2195  
11  
10 NO FPP INSTRUCTIONS GENERATED  
9  
8 .T117.(320,40)A,B/-SP=SYO.(320,40)HDRHIS  
7

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14.22.52    30-OCT-80      PAGE 1
HIST.FTN      /TR:BLOCKS/WR

0001      SUBROUTINE HIST(BUF, STARTP, STOPP, RCHAN, TST)
C
C      THIS ROUTINE WILL BUILD HISTOGRAMS FOR EACH CHANNEL USING
C      THE VALUE FROM THE FULL FRAME DATA
C
0002      IMPLICIT INTEGER (A-Z)
0003      REAL H4, H5, H6, H7
0004      LOGICAL*1 BUF(1), MISNO(2)
0005      COMMON/HIST/H4(256), H5(256), H6(256), H7(256)
0006      COMMON/LABEL/NAME(3), MISNO, E
C
C      ROUTINE ID
C
0007      NAME(1)='HI'
0008      NAME(2)='ST'
0009      NAME(3)=' '
C
C      BUILDING HISTOGRAM FOR EACH CHANNEL
C
0010      GOTO(10, 20, 30, 40)RCHAN
C
0011      10 CONTINUE
0012      K=0
0013      DO 15 M=STARTP, STOPP
0014      K=K+1
0015      N=STARTP-TST+K
0016      PIXEL=BUF(N).AND."377
0017      H4(PIXEL+1)=H4(PIXEL+1)+1
0018      15 CONTINUE
0019      GOTO 9999
C
0020      20 CONTINUE
0021      K=0
0022      DO 25 M=STARTP, STOPP
0023      K=K+1
0024      N=STARTP-TST+K
0025      PIXEL=BUF(N).AND."377
0026      H5(PIXEL+1)=H5(PIXEL+1)+1
0027      25 CONTINUE
0028      GOTO 9999
C
0029      30 CONTINUE
0030      K=0
0031      DO 35 M=STARTP, STOPP
0032      K=K+1
0033      N=STARTP-TST+K
0034      PIXEL=BUF(N).AND."377
0035      H6(PIXEL+1)=H6(PIXEL+1)+1
12 0036      35 CONTINUE
11 0037      GOTO 9999
C
10 0038      40 CONTINUE
9 0039      K=0
8 0040      DO 45 M=STARTP, STOPP
7 0041      K=K+1
6 0042      N=STARTP-TST+K

```

ORIGINAL PRINTING  
OF POOR QUALITY

```
FORTRAN IV-PLUS V02-51E      14 22 52      30-OCT-80      PAGE 2
HIST. PTN      /TR: BLOCKS/WR

0043      PIXEL=BUF(N). AND. "377
0044      H7(PIXEL+1)=H7(PIXEL+1)+1
0045      45 CONTINUE
          C
0046      9999 RETURN
0047      END
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTAN IV-PLUS V02-51E 14:22:52 30-OCT-80 PAGE 3  
HIST.FTN /TR.BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000570 220	RM. I. CON. LCL
2	\$PDATA	000012 5	RM. D. CON. LCL
3	\$IDATA	000012 5	RM. D. CON. LCL
4	\$VARS	000010 4	RM. D. CON. LCL
5	\$TEMPS	000004 2	RM. D. CON. LCL
6	HIST	010000 2048	RM. D. OVR. CBL
7	LABEL	000012 5	RM. D. TVR. CBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
HIST		1-000000									

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
E	I*2	7-000010	K	I*2	4-000000	M	I*2	4-000002	N	I*2	4-000004
KCHAN	I*2	F-000010*	STARTP	I*2	F-000004*	STDP	I*2	F-000006*	TST	I*2	F-000012*

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
BUF	L*1	F-000002*	000001	0 (1)
H4	R*4	6-000000	002000	512 (256)
H5	R*4	6-000000	002000	512 (256)
H6	R*4	6-000000	002000	512 (256)
H7	R*4	6-000000	002000	512 (256)
HIST0	L*1	7-000006	000002	1 (2)
NAME	I*2	7-000000	000006	3 (3)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000076	15	**	20	1-000236	25	**
35	**	40	1-000532	45	**	9999	1-000666

TOTAL SPACE ALLOCATED = 010742 2289

IT17 (320, 40)A B/-SP=SYO: (320, 40)HIST

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTAN IV-PLUS V02-51E		14:25:12	30-OCT-80	PAGE 1
PARTP FTM		/TR: BLOCKS/WR		
0001	SUBROUTINE PARTP(PTCH, STARTP, STOPP, STOP, STRIPC, TSE, TST, PIXELS, + I, PATCH, PSIZE)			
C				
C	THIS ROUTINE WILL ADJUST THE PARAMETERS IF A PATCH IS DIVIDED BETWEEN			
C	TWO CCT STRIPS.			
C				
C	LOCAL VARIABLES:			
C				
C	CCTNO= THE CCT STRIP NUMBER LAST STORED			
C	PIECE= AN ARRAY CONTAINING THE PARAMETER VALUES TO USE FOR THE			
C	REMAINDER OF THE RIGHT MOST PATCH ON NEXT CCT STRIP			
C				
0002	IMPLICIT INTEGER (A-Z)			
0003	LOGICAL PTCH			
0004	DIMENSION PIECE(2,4)			
0005	DATA CCTNO/0/, PIECE/8*0/			
C				
0006	IF(STRIPC.EQ.1.AND.STOP.GT.TSE)GOTO 20			
0007	IF(STRIPC.EQ.1)GOTO 9999			
0008	IF(.NOT.PTCH)GOTO 10			
0009	IF(PIECE(STRIPC-1,2).NE.PATCH-1)GOTO 10			
C				
C	THE CURRENT PATCH IS THE PORTION OF THE UNPROCESSED RIGHT MOST PATCH			
C	FROM PREVIOUS CCT STRIP			
C				
0010	PTCH=.FALSE.			
0011	STARTP=TST			
0012	STOPP=TST+(PSIZE-PIECE(STRIPC-1,1))-1			
0013	NPIXEL=(PIECE(STRIPC-1,2)-1)*PSIZE+(70+512*(I-1))			
0014	PIXELS=NPIXEL+PIECE(STRIPC-1,1)			
0015	PATCH=PATCH-1			
C				
0016	10 CONTINUE			
0017	IF(STOP.LE.TSE)GOTO 9999			
C				
0018	20 CONTINUE			
0019	IF(CCTNO.EQ.STRIPC)GOTO 9999			
C				
C	SAVE THE NUMBER OF PIXELS PROCESSED FOR THE RIGHT MOST PATCH ON			
C	CURRENT CCT STRIP			
C				
0020	PIECE(STRIPC,1)=(STOPP-STARTP)+1			
0021	PIECE(STRIPC,2)=PATCH			
0022	CCTNO=STRIPC			
C				
0023	9999 RETURN			
0024	END			

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14.25.12 30-OCT-80 PAGE 2  
PARTP. FTN /TR: BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000322	105 RW, I, CON, LCL
4	\$VARS	000024	10 RW, D, CON, LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
PARTP		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CCTND	I*2	4-000020	I	I*2	F-000022	NP1XEL	I*2	4-000022
P1XEL	I*2	F-000022	P1XEL	I*2	4-000022	PATCH	I*2	F-000024
STRIPC	I*2	F-000012	TSE	I*2	F-000014	TST	I*2	F-000016
						STOP	I*2	F-000010
						STOPP	I*2	F-000006

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
PIECE	I*2	4-000000	000020	8 (2,4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000224	20	1-000242	9999	1-000320

TOTAL SPACE ALLOCATED = 000346 115

NO FPP INSTRUCTIONS GENERATED

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E

14:25.15

30-OCT-80

PAGE 3

PARTP. FTN

/TR: BLOCKS/WR

, TT17: [320, 40]A. 8/-SP=SYO: [320, 40]PARTP



ORIGINAL PAGE 15  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14 26 10	30-OCT-80	PAGE 1
INTOCT.FTN		/TR:BLOCKS/WR		
0001		SUBROUTINE INTOCT(INTEG,OCTAL)		
	C			
	C	THIS PROGRAM WILL CONVERT A DECIMAL NUMBER TO OCTAL		
	C	IT CAN HANDLE VALUES IN DECIMAL FROM 1 TO 4095		
	C			
	C	DECIMAL NUMBER LESS THAN 4096		
	C			
0002		INTEGER OCTAL		
0003		IF(INTEG.LT.512)GO TO 10		
0004		K=MOD(INTEG,512)		
0005		L=MOD(K,64)		
0006		M=MOD(L,8)		
0007		IOCT=INTEG/512*1000+K/64*100+L/8*10+M		
0008		GO TO 100		
	C			
	C	DECIMAL NUMBER LESS THAN 512 GE 64		
	C			
0009	10	IF(INTEG.LT.64)GO TO 20		
0010		K=MOD(INTEG,64)		
0011		L=MOD(K,8)		
0012		IOCT=INTEG/64*100+K/8*10+L		
0013		GO TO 100		
	C			
	C	DECIMAL NUMBER LT 64 GE 8		
	C			
0014	20	IF(INTEG.LT.8)GO TO 30		
0015		K=MOD(INTEG,8)		
0016		IOCT=INTEG/8*10+K		
0017		GO TO 100		
	C			
	C	DECIMAL NUMBER LT 8		
	C			
0018	30	IOCT=INTEG		
0019	100	OCTAL=IOCT		
0020		RETURN		
0021		END		

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:26:10 30-OCT-80 PAGE 2  
INTCT.FIN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000432 141	RW, I, CON, LCL
4	\$VARS	000010 4	RW, D, CON, LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
INTCT		1-000000									

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
INTEG	I*2	F-000002*	IDCT	I*2	4-000000	K	I*2	4-000000	L	I*2	4-000002
OCTAL	I*2	F-000004*							M	I*2	4-000004

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000172	20	1-000314	30	1-000400	100	1-000414

TOTAL SPACE ALLOCATED = 000442 145

NO FPP INSTRUCTIONS GENERATED

1117: (320, 401A, B/-SP-SV0: (320, 40)INTCT



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E  
UNVRSL. FTM /TR: BLOCKS/WR

14:26:27 30-OCT-80

PAGE 2

	C	PLACE THE MISSION NUMBER INTO HEADER
	C	
0030		UHEADR(66)=E
	C	
	C	PLACE ACTIVE CHANNEL FLAGS; 1=ACTIVE, 0=NOT ACTIVE
	C	
0031		UHEADR(81)=240
0032		UHEADR(90)=4
	C	
	C	PLACE PIXEL SIZE
	C	
0033		UHEADR(91)=8
	C	
	C	PLACE VIDEO DATA START, INDICATING DATA FOLLOWS HEADER
	C	
0034		UHEADR(93)=1
	C	
	C	PLACE PIXELS PER LINE PER BAND
	C	
0035		UHEADR(96)=2
0036		UHEADR(97)=0
	C	
	C	PHYSICAL RECORD SIZE; MULTIPLE OF 180
	C	ACTUAL SIZE IS (16*32*4=2048 BYTES)
	C	
0037		PIXLN=2560
0038		UHEADR(100)=BWORD(2)
0039		UHEADR(101)=BWORD(1)
	C	
	C	PLACE THE DATA SET SIZE; NUMBER OF RECORDS PER LINE
	C	
0040		UHEADR(104)=1
	C	
	C	ANCILLARY BLOCK SIZE
	C	
0041		UHEADR(106)=70
	C	
	C	BINARY START PIXEL
	C	
0042		UHEADR(109)=1
	C	
	C	BINARY STOP PIXEL
	C	
0043		UHEADR(110)=2
0044		UHEADR(111)=0
	C	
	C	GENERATION COMPUTER WORD SIZE
	C	
12 0045		UHEADR(1753)=16
11	C	
10	C	DATA SETS PER PHYSICAL RECORD
9	C	
8 0046		UHEADR(1778)=1
7	C	
6	C	BANDS IN PRIMARY RECORD
	C	

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14 26 27	30-OCT-80	PAGE 3
UNVRSL FTM		/TR: BLOCKS/WR		
0047	UHEADR(1786)=4			
	C			
	C BYTES PER LINE PER BAND			
	C			
0048	UHEADR(1787)=2			
0049	UHEADR(1788)=0			
	C			
	C PIXEL SKIP FACTOR			
	C			
0050	UHEADR(1790)=1			
	C			
	C LINE SKIP FACTOR			
	C			
0051	UHEADR(1792)=1			
	C			
	C MISSION NUMBER			
	C			
0052	UHEADR(1817)=E			
	C			
	C LINES PER BAND			
	C			
0053	UHEADR(1819)=2			
0054	UHEADR(1820)=0			
	C			
	C PIXELS PER LINE PER BAND			
	C			
0055	UHEADR(1821)=2			
0056	UHEADR(1822)=0			
	C			
	C LINE SKIP FACTOR			
	C			
0057	UHEADR(1823)=1			
	C			
	C PIXEL SKIP FACTOR			
	C			
0058	UHEADR(1824)=1			
	C			
	C ACTIVE BAND FLAGS			
	C			
0059	UHEADR(1825)="360			
	C			
	C CLOUD COVER			
	C			
0060	UHEADR(1827)=1			
	C			
	C INGEST BAND STATUS			
	C			
0061	UHEADR(1828)=2			
0062	UHEADR(1829)=2			
0063	UHEADR(1830)=2			
0064	UHEADR(1831)=2			
	C			
	C SUN ELEVATION ANGLE			
	C			
0065	SUNELA=SUNEL			
0066	UHEADR(1833)=B(2)			

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E          14:26:27          30-OCT-80          PAGE 4
UNVRSL. FTM          /TR: BLOCKS/WR

0067          UHEADR(1834)=B(1)
C
C          LATITUDE; CENTER OF FULL FRAME
C
0068          ILAT=FLAT
0069          DLAT=FLAT
0070          RLAT=((ILAT-DLAT)*100)/60
0071          UHEADR(1985)=LATD
0072          UHEADR(1986)=RWLAT(2)
0073          UHEADR(1987)=BWLAT(1)
0074          UHEADR(1989)=RLAT
C
C          LONGITUDE; CENTER OF FULL FRAME
C
0075          ILONG=FLONG
0076          DLONG=FLONG
0077          RLONG=((ILONG-DLONG)*100)/60
0078          UHEADR(1990)=LOND
0079          UHEADR(1991)=BWLONG(2)
0080          UHEADR(1992)=BWLONG(1)
0081          UHEADR(1994)=RLONG
C
C          WRITE THE BUFFER TO THE HEADER FILE
C
0082          WRITE(LUN2,'2')UHEADR
C
0083          RETURN
0084          END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

1 FORTTRAN IV-PLUS V02-51E 14:26:27 30-OCT-80 PAGE 5  
UNVRSI FTR BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1 001054	279	RM.I. CON. LCL
3	\$IDATA 000034	14	RM.D. CON. LCL
4	\$VARS 006124	1578	RM.D. CON. LCL
6	LABEL 000012	5	RM.D. OVR. CBL
7	LATLON 000004	2	RM.D. OVR. GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
UNVRSI		1-000000									

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
DLAT	I*2	4-000002	DLONG	I*2	4-000000	E	I*2	6-000010	FLONG	R*4	F-000010*
I	I*2	4-006122	ID	I*2	4-006112	ILAT	R*4	4-000010	IM	I*2	4-006110
IV	I*2	4-006114	J	I*2	4-006120	JDAY	I*2	4-006116	K	I*2	4-006106
LONGD	I*2	7-000002	LUN2	I*2	F-000002*	PIXLN	I*2	4-000006	RLAT	R*4	4-000020
SUNEL	I*2	F-000004*	SUNELA	I*2	4-000004				RLONG	R*4	4-000024

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
ASCLIV L*1	4-006103	000002	1	(2)
ASCLJD L*1	4-006100	000003	1	(3)
B	L*1	4-000004	000002	1 (2)
BMLAT L*1	4-000002	000002	1	(2)
BMLONG L*1	4-000000	000002	1	(2)
BWDRD L*1	4-000006	000002	1	(2)
ITIME L*1	4-006070	000010	4	(8)
MISNO L*1	6-000006	000002	1	(2)
NAME	I*2	6-000000	000006	3 (3)
TFORM L*1	4-006030	000040	16	(32)
UNEADR L*1	4-000030	006000	1536	(3072)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	3-000000	2	3-000004	5	**	10	**

FUNCTIONS AND SUBROUTINES REFERENCED

IRATE	JULIAN	TIME
-------	--------	------

FORTRAN IV-PLUS V02-51E 14 26:27 30-OCT-80 PAGE 6  
UNVRSL. FTN /TR: BLOCKS/WR

TT17: (320, 40)A. B/-SP=SYO: (320, 40)UNVRSL



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E	14 28 47	30-OCT-80	PAGE 1
FRONT FTM	/TR BLOCKS/WR		
C*****			
C	LOCKHEED ELECTRONICS COMPANY, INC.		
C	REMOTE SENSING APPLICATIONS LAB		
C	( R S A L )		
C	NAME: LEFT JUSTIFY ASCII CHARACTER STRING (FRONT)		
C	PROCURED FROM: NASA, JOHNSON SPACE CENTER		
C	BLDG. 17, DATA TECHNIQUES LAB		
C	ON: 01-MAY-78		
C	REVISED BY:	ON:	PURPOSE FOR REVISION:
C	XXXXXXXXXX	XX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
C	B. L. TAYLOR	22-MAY-78	BRING TO RSAL STANDARD
C	PROGRAM DESCRIPTION:		
C	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		
C	LEFT JUSTIFIES AN ASCII CHARACTER STRING REMOVES ANY LEADING		
C	BLANKS AND BLANK FILLS THE ARRAY AFTER THE LAST CHARACTER IS		
C	SHIFTED.		
C	ENTRY POINTS:		
C	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		
C	FRONT		
C	PROGRAM INITIATION OR CALL SEQUENCE:		
C	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		
C	CALL FRONT(IARRAY, NUM)		
C	CALL ARGUMENTS:	FORMATS:	
C	XXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
C	IARRAY	BYTE STRING (ARRAY) TO BE LEFT JUSTIFIED.	
C	NUM	MAXIMUM NUMBER OF BYTES IN THE STRING (ARRAY).	
C	INPUTS:	FORMATS:	
C	XXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
C	NONE		
C	OUTPUTS:	FORMATS:	
C	XXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
C	NONE		
C	SPECIAL REMARKS:		
C	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		
C	NONE		

•

```

FORTRAN IV-PLUS V02-51E          14:28:47      30-OCT-80          PAGE 2
FRONT.FTN           /TR:BLOCKS/WR

C
C   ROUTINES CALLED:
C   <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C   NONE
C
C
C
C-----
C

C LEFT JUSTIFY A CHARACTER STRING
0001      SUBROUTINE FRONT(I,N)
0002      BYTE I
0003      DIMENSION I(N)
C
0004      DO 1 J=1,N
0005      IF(I(J).NE."40")GO TO 2
0006      1    CONTINUE
0007      RETURN
C
0008      2    K=1
0009      DO 3 J=J,N
0010      I(K)=I(J)
0011      3    K=K+1
0012      IF(K.GT.N)RETURN
C
0013      DO 4 J=K,N
0014      I(J)="40"
0015      RETURN
C
0016      END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```
1  FORTRAN IV-PLUS V02-51E      14:26:47  30-OCT-80      PAGE 3
   FORT. FTN      /TR: BLOCKS/MR

PROGRAM SECTIONS

NUMBER  NAME      SIZE      ATTRIBUTES
-----  -
1  %CODE1  000264    90      RM. I. CON. LCL
3  %DATA   000020     8      RM. D. CON. LCL
4  %VARS   000004     2      RM. D. CON. LCL
5  %EMPS   000002     1      RM. D. CON. LCL

ENTRY POINTS

NAME  TYPE  ADDRESS  NAME  TYPE  ADDRESS  NAME  TYPE  ADDRESS
-----
FRONT 1-000000

VARIABLES

NAME  TYPE  ADDRESS  NAME  TYPE  ADDRESS  NAME  TYPE  ADDRESS
-----
J    I*2  4-000000  K    I*2  4-000002  N    I*2  F-000004*

ARRAYS

NAME  TYPE  ADDRESS  SIZE  DIMENSIONS
-----
I    L*1  F-000002*  **    (*)

LABELS

LABEL  ADDRESS  LABEL  ADDRESS  LABEL  ADDRESS  LABEL  ADDRESS
-----
1      **      2      1-000114  3      **      4      **

TOTAL SPACE ALLOCATED = 000312  101

NO FPP INSTRUCTIONS GENERATED

.TT17 (320.40JA.B7-SP-SY0 (320.40JFRONT
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTAN IV-PLUS V02-51E  
JULIAN.FTN /TR: BLOCKS/WR

14:28:33

30-OCT-80

PAGE 1

C\*\*\*\*\*

C

C

C

LOCKHEED ELECTRONICS COMPANY, INC.  
REMOTE SENSING APPLICATIONS LAB.  
( R S A L )

C

C

C

C

C

DEVELOPED BY: TED KELL  
ON:

C

C

C

C

REVISED BY:	ON:	PURPOSE FOR REVISION:
XXXXXXXXXX	XX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
S OBRIEN	6/78	ADD COMMENTS FOR STANDARDIZATION

C

C

C

C

C

C

C

PROGRAM DESCRIPTION:  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX  
THIS FUNCTION COMPUTES JULIAN DAY INTO INTEGER FORMAT

C

C

C

C

C

C

C

C

C

ENTRY POINTS:  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX  
NONE

C

C

C

C

C

C

PROGRAM INITIATION OR CALL SEQUENCE:  
XXXXXXXXXXXXXXXXXXXXXXXXXXXX  
IVAR=JULIAN(MON, IDAY, IYR)

C

C

C

C

C

CALL ARGUMENTS:	FORMATS:
XXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
MON	MONTH NUMBER IN INTEGER
IDAY	DAY NUMBER IN INTEGER
IYR	YEAR NUMBER IN INTEGER

C

C

C

C

C

INPUTS:	FORMATS:
XXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX
MON	SEE ABOVE

C

C

C

C

C

C

OUTPUTS:	FORMATS:
XXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX

C

C

**ORIGINAL PAGE IS  
OF POOR QUALITY**

**14: 28-33**

**PAGE 2**

**C**

JULIAN DAY NUMBER IN INTEGER

**C**

**C**

**C**

**C**

**SPECIAL REMARKS:**

**C**

**C**

**C**

**C**

**20**

**ROUTINES CALLED:**

**C**

**C**

**NONE**

**C**

**c**

$$\frac{C}{2}$$

cc

$$\frac{c}{c}$$

CC

**C**

**C**

C-\*\*\*\*\*

```
0001      INTEGER FUNCTION JULIAN(M,D,Y)
```

0002	INTEGER M, D, Y, S(12)
------	------------------------

0003 DATA S/O, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334/

```
0004      IF (M. NE. 0. AND. D. NE. 0) GO TO 1
```

0005 JULIAN = 0

0006 RETURN

0007 1 CONTINUE

0008 JULIAN=S(M)+D

```
0009      IF (MO. LE. 2) RETURN
```

```
0010      IF ((Y/4)*4. EQ. Y) JULIAN=JULIAN+1
```

0011 RETURN

0012 END

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTAN IV-PLUS V02-51E 14:28:33 30-OCT-80 PAGE 3  
JULIAN FTY /TR: BLOCKS/WR

# PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	SCODE1	000144	50 RM. I. CC. A. LCL
3	SIDATA	000002	1 RM. D. CON. LCL
4	SIVARS	000032	13 RM. D. CON. LCL

# ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
JULIAN	I*2	1-000000						

# VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
D	I*2	F-000004*	M	I*2	F-000002*	M0	I*2	4-000030
						V	I*2	F-000006*

# ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
S	I*2	4-000000	000030	12 (12)

# LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000050				

TOTAL SPACE ALLOCATED = 000200 64

NO FPP INSTRUCTIONS GENERATED

.I117: [320.40]A. B/-SP=SY0: [320.40]JULIAN



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E

14:28:16

30-OCT-80

PAGE 2

HROS. FTN /TR:BLOCKS/WR

C-\*\*\*\*\*  
C

```

0001 SUBROUTINE HROS(LUN1,HEADER,FMT,EOF,PTY,IPRM)
0002 IMPLICIT INTEGER(A-Z)
0003 INTEGER HCOMB(20)
0004 DIMENSION ISTAT(2),IPRM(6)
0005 BYTE HEADER(1),SCNID(12),MISNO(2),ISTATB(2)
0006 COMMON/HCOM/SS,SE,LS,LE,NRPDS,NDSPR,NCPR,NRPC,ANCL,NC,NS,
1 NBIT,DOI,NCAR,SVD,RSIZ,PSKIP,HSIZ,CALP,CERR,NSPR
0007 COMMON/LABEL/NAME(3),MISNO,E
0008 EQUIVALENCE(SS,HCOMB(1)),(ISTAT(1),ISTATB)
0009 DATA RLB/"1000/",IEEOF/-10/
0010 BBUF(1)=IBYTE(0,HEADER(1))*256+IBYTE(0,HEADER(1+1))
0011 NAME(1)='HP'
0012 NAME(2)='RO'
0013 NAME(3)='S'
0014 IF(CERR.NE.0)RETURN
0015 IF(FMT.EQ.1) GO TO 1 !FORMAT=UNIVERSAL
0016 IF(FMT.EQ.2) GO TO 2 !FORMAT=LANDSAT
0017 IF(FMT.EQ.3) GO TO 3 !FORMAT=LARSYS
0018 CERR = 9
0019 RETURN

```

C BUILD HCOM FOR UNIVERSAL FORMAT TAPE

```

0020 1 SS = BBUF(108)
0021 SE=BBUF(110)
0022 LE = 0
0023 NRPDS=HEADER(104)
0024 NDSPR=HEADER(1778)
0025 NCPR=HEADER(102)
0026 NRPC=HEADER(103)
0027 ANCL=BBUF(105)
0028 NC=HEADER(90)
0029 NS=BBUF(1787)
0030 NBIT=HEADER(91)
0031 DOI = HEADER(107)
0032 NCAR=BBUF(1785)
0033 SVD = BBUF(92)
0034 IF(SVD.EQ.0)SVD=1
0035 RSIZ = BBUF(100)
0036 PSKIP = 0
0037 HSIZ = 1530
0038 CALP = 0
0039 NPSL=B2I(HEADER,96,97)
0040 GOTO 9999

```

C  
C BUILD HCOM FOR LARSYS FORMAT TAPE

```

12 C
11 0041 3 SS = 1
10 0042 SE = BBUF(23)
9 0043 LE = BBUF(39)
8 0044 NRPDS = 1
7 0045 NDSPR = 1
6 0046 NCPR = BBUF(19)
5 0047 NRPC = 1

```



```

FORTRAN IV-PLUS V02-51E          14:28:16          30-OCT-80          PAGE 3
HPRS. FTN          /TR: BLOCKS/WR

0048          ANCL = 4
0049          NC = NCPR
0050          NS = SE
0051          NBIT = 8
0052          DOI = 0
0053          NCAR = NCPR
0054          SVD = 1
0055          RSIZ = NC*NS+4
0056          NS=NS-4
0057          SE=SE-4
0058          PSKIP = 0
0059          HSIZ = 400
0060          CALP = 6
0061          GOTO 9999

C
C          BUILD HCOM FOR LANDSAT FORMAT TAPE
C
0062          2          NS = BBUF(39)/4
0063                      I=HEADER(14)
0064                      SS = NS*(IAND(I,15)-1)+1
0065                      SE = SS+NS-1
0066                      LS = 1
0067                      LE = 2340
0068                      NRPDS = 1
0069                      NDSPR = 1
0070                      NCPR = 4
0071                      NRPC = 1
0072                      ANCL = 0
0073                      NC = 4
0074                      NBIT = 8
0075                      DOI = 2
0076                      NCAR = 4
0077                      SVD = 1
0078                      RSIZ = BBUF(17)
0079                      PSKIP = 9
0080                      HSIZ = 338
0081                      CALP = 0

C
0082          4 CONTINUE
C
C          READ ANCILLARY RECORD & CONCATENATE TO ID RECORD
C          TO FORM HEADER RECORD FOR LANDSAT TAPE
C
C          PUTTING STARTING ADDRESS OF HEADER ARRAY INTO IPRM ARRAY
C
C
C          READING FROM TAPE
C
0083                      IPRM(1)=IPRM(1)+40
0084                      IPRM(2)=624
0085                      CALL WTGIO(RLB,LUN1,1,,ISTAT,IPRM,ISW)

C
C          ERROR CHECK
C
0086          5 IF(ISTATB(1).GE."000")GO TO 9999          'READ OKAY="001
0087                      IF(ISTATB(1).EQ."EOF")EOF=1          'END OF FILE="366"--10

```

ORIGINAL TEST  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:28:16 30-OCT-80 PAGE 4  
HROS.FTN /TR:BLOCKS/WR

0088 IF(EOF.EQ.1)RETURN  
0089 IF(ISTATB(1).EQ."374")PC=PC+1 !PARITY CHECK="374"--4  
0090 IF(ISTATB(1).EQ."374")GO TO 4  
0091 CERR=ISTATB(1) !SET READ ERROR CODE  
0092 9999 RETURN  
0093 END

NUMBER	NAME	SIZE	ATTRIBUTES
--------	------	------	------------

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

[illegible]

HPRODS 1-000000

[illegible]

SSUF I#2 1-001372

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ANCL	1:2	6-000020	CALP	1:2	6-000044	CLRR	1:2	6-000046	D01	1:2	6-000030
LCIF	1:2	6-000010*	FHF	1:2	F-000008*	HS1Z	1:2	6-000042	I	1:2	F-000002*
LEUF	1:2	4-000022	LSW	1:2	4-000030	LE	1:2	6-000006	LS	1:2	6-000032
NS1T	1:2	6-000026	NC	1:2	6-000032	NCR	1:2	6-000032	NCSR	1:2	6-000014
NS2L	1:2	4-000024	NRPC	1:2	6-000016	NRDPS	1:2	6-000010	NS	1:2	6-000024
PC	1:2	4-000032	PRTY	1:2	4-000012*	PSKP	1:2	6-000040	RLB	1:2	4-000020
SP	1:2	6-000002	SS	1:2	6-000000	SV	1:2	6-000034	RS1Z	1:2	6-000036

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35				
36				
37				
38				
39				
40				
41				
42				
43				
44				
45				
46				
47				
48				
49				
50				
51				
52				
53				
54				
55				
56				
57				
58				
59				
60				
61				
62				
63				
64				
65				
66				
67				
68				
69				
70				
71				
72				
73				
74				
75				
76				
77				
78				
79				
80				
81				
82				
83				
84				
85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95				
96				
97				
98				
99				
100				

NAME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
NAME	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100

5731-1



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E

14:27:54

30-OCT-80

PAGE 1

HDREAD.FTN /TR:BLOCKS/WR

C\*\*\*\*\*

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

LOCKHEED ELECTRONICS COMPANY, INC.  
REMOTE SENSING APPLICATIONS LAB.  
( R S A L )

NAME: HDREAD

DEVELOPED BY: SUSANNE OBRIEN  
ON: 6/78

REVISED BY: ON: PURPOSE FOR REVISION:

\*\*\*\*\*

PROGRAM DESCRIPTION:

\*\*\*\*\*

THIS ROUTINE WILL READ HEADER DATA RECORDS FROM A FOREIGN  
IMAGERY TAPE IN ONE OF THE FOLLOWING FORMATS: UNIVERSAL,  
LANDSAT 1 OR 2, LARSYS 2 OR 3

ENTRY POINTS:

\*\*\*\*\*

NONE

PROGRAM INITIATION OR CALL SEQUENCE:

\*\*\*\*\*

CALL HDREAD(LUN, IFRM, HEADER, BFSZ, EOF, PC, IE, BUFFER)

CALL ARGUMENTS:

FORMATS:

\*\*\*\*\*

\*\*\*\*\*

LUN

I1 LOGICAL UNIT NUMBER

IFRM

01 FORMAT OF IMAGE TAPE

1=UNIVERSAL

2=LANDSAT

3=LARSYS

0=ERROR HEADER RECORD NOT  
ACCEPTABLE SIZE

HEADER

BYTE STRING BUFFER FOR HEADER DATA

BFSZ

I1 SIZE OF READ IN BUFFER 'BUFFER'

EOF

I1 END OF FILE 1=END

PC

PARITY COUNT

IE

01 ERROR CODE

IE=0 MEANS NO ERROR

BUFFER

BYTE STRING BUFFER TO READ DATA INTO

INPUTS:

FORMATS:

\*\*\*\*\*

\*\*\*\*\*

OUTPUTS:

FORMATS:

\*\*\*\*\*

\*\*\*\*\*

HCOM

COMMON DATA AREA BUILT BY HPROS (CALLED)

SPECIAL REMARKS:

\*\*\*\*\*

ORIGINAL PAGE  
OF PCR QUALITY

```

FORTRAN IV-PLUS V02-51E          14-27-84      30-OCT-80      PAGE 2
HREAD.FTN       /TR:BLOCKS/WR

C
C   ROUTINES CALLED:
C   <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
C   MTHREAD           MAG TAPE RECORD READ
C   HPROS            HEADER PROCESS TO BUILD HCOM COMMON AREA
C   IBYTE             CONVERTS DATA FROM ASCII TO INTEGER
C
C-----
C

0001      SUBROUTINE HREAD(LUN1, IFRM, HEADER, BFSZ, EOF, PC, IE, BUFFER, INS, HDRADR,
+         ISS)
0002      IMPLICIT INTEGER (A-Z)
0003      BYTE HEADER(1), ISTATB(2)
0004      BYTE BUFFER(1), SCNID(12), MISNO(2)
0005      DIMENSION ISTAT(2), IPRM(6), IPRM2(6)
0006      EQUIVALENCE (ISTAT(1), ISTATB)
0007      COMMON /HCOM/SS, SE, LS, LE, NRPDS, NDSPR, NCPR, NRPC, ANCL, NC, NS,
        1 NBIT, DOI, NCAR, SVD, RSIZ, PSKIP, HSIZ, CALP, CERR, NSPR
0008      COMMON/LABEL/NAME(3), MISNO, E
0009      INTEGER LSLOC(3)
0010      DATA LSLOC/71, 0, 1/, RLB/"1000/", IPRM(2)/3060/

C
0011      NAME(1)='HD'
0012      NAME(2)='RE'
0013      NAME(3)='AD'
0014      ISS=SS
0015      EOF=0
0016      IEEEOF=-10

C
C   PUTTING STARTING ADDRESS OF HEADER INTO IPRM ARRAY
C
0017      IPRM(1)=HDRADR

C
C   READING HEADER FROM TAPE
C
0018      CALL WTQIP(RLB, LUN1, 1., ISTAT, IPRM, ISW)

C
C   ERROR CHECK
C
0019      CERR=0
0020      IFRM=0
0021      IF(ISTATB(1).GE."000")GO TO 20      !READ OKAY?
0022      IF(ISTATB(1).EQ.IEEOF)EOF=1          !END OF FILE="366=-10
0023      IF(EOF.EQ.1)RETURN
0024      IF(ISTATB(1).EQ.-4)PC=PC+1           !PARITY CHECK="374=-4
0025      IF(ISTATB(1).EQ.-4)GO TO 20
0026      GO TO 110                            !READ ERROR
0027      IF(ISTAT(2).EQ.3060)IFRM=1           !UNIVERSAL FORMAT
0028      IF(ISTAT(2).GE.40.AND.ISTAT(2).LE.42)IFRM=2    !LANDSAT FORMAT
0029      IF(ISTAT(2).GE.800.AND.ISTAT(2).LE.902)IFRM=3  !LARSYS FORMAT
0030      IF(IFRM.NE.0)GO TO 30
0031      CERR = ISTAT(2) !FORMAT      ERROR
0032      GO TO 100

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14 27 54      30-OCT-80      PAGE 3
HOREAD.FTN      /TR:BLOCKS/WR

0033      30      CALL HPROS(LUN1,HEADER,IFRM,EOF,PC,IPRM)      'BUILD HCOM
0034      IF(CERR.NE.0)GO TO 100
0035      INS=NS
C          THIS ROUTINE FOR MULTIPLE RECS PER CHANNEL
0036      IF(NRPC.LE.1)GO TO 40
0037      NSPR=NS/NRPC      !CALC #PIX PER REC
0038      IF(MOD(NS,NRPC).EQ.0)GO TO 40      !EVEN DIVISION GO TO 4
0039      NSPR=NSPR+1
0040      40      Q = RSIZ
0041      IF(RSIZ.GT.DFSZ) CERR=12
0042      IF(CERR.EQ.12)GO TO 100 !REC TOO BIG FOR BUFFER
0043      IPRM2(2)=RSIZ
C
C          CALL HCODE FOR HEADER DECODE OF BUFFER
C
0044      CALL HCODE(IFRM,HEADER,IE,LVL)
C
C          PUTTING STARTING ADDRESS OF BUFFER INTO IPRM2 ARRAY
C
0045      CALL GETADR(IPRM2,BUFFER)
C
C          READING IMAGE DATA FROM TAPE
C
0046      CALL WTGIO(RLD,LUN1,1,,ISTAT,IPRM2,ISW)
C
C          ERROR CHECK
C
0047      45 IF(ISTATB(1).GE."000")GOTO 50      !READ OKAY=1
0048      IF(ISTATB(1).EQ.IEEOF)EOF=1      !END OF FILE?
0049      IF(EOF.EQ.1)RETURN
0050      IF(ISTATB(1).EQ.-4)PC=PC+1      !PARITY CHECK?
0051      IF(ISTATB(1).EQ.-4)GO TO 50
0052      GO TO 110      !READ ERROR
0053      50      IF(IFRM.EQ.2)GO TO 60      !LANDSAT TAPE
C          CALCULATE LINE START NUMBER
0054      LS=LSLOC(IFRM)
0055      LS=IBYTE(0,BUFFER(LS))*256+IBYTE(0,BUFFER(LS+1))
C          VERIFY DATA
C          CHECK DATA ORDER INDICATOR
0056      60      IF(DOI.EQ.0 OR.(IFRM.EQ.2 AND.DOI.EQ.2)) GO TO 70
0057      CERR =15
0058      GO TO 100
0059      70      IF(NBIT.EQ.8) GO TO 80      !CHECK # BITS PER BYTE
0060      CERR=16
0061      GO TO 100
0062      80      IF(NRPC.GE.1) GO TO 90      !# RECS PER CHAN
0063      NRPC=1
0064      NCPR=0      !# CHANS PER REC
C          CHECK # RECS PER DATA SET AND RESET NCPR IF NEEDED
0065      IF(NRPDS.GT.1) NCPR=(NC-NCAR)/(NRPDS-1)
0066      90      IF(NDSPR.GE.1)GO TO 100      !# DATA SETS PER REC
0067      NDSPR=1 !NDSPR CANNOT =0 IN CALCS
0068      100      IE=CERR      !SET ERROR CODE
0069      RETURN
0070      110      IE=ISTATB(1)      !SET READ ERROR CODE
0071      9999 RETURN

```

ORIGINAL PAGE 1  
OF FOUR

FORTAN IV-PLUS V02-51E 14 27 54 30-OCT-80 PAGE 4  
HDREAD.FTN /TR: BLOCKS/WR

0072 END



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14.27.54 30-OCT-80 PAGE 5  
HREAD FIN 7TR BLOCKS/MR

# PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	SCUDCI	001402	385 RM. I. CON. LCL
2	SPDATA	000010	4 RM. D. CON. LCL
3	STDATA	000136	47 RM. D. CON. LCL
4	SWARS	000070	28 RM. D. CON. LCL
5	HUM	000052	21 RM. D. OVR. GBL
6	LABL	000012	5 RM. D. OVR. GBL

## ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
HREAD		1-000000						

## VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ARCL	1-2	6-000020	BFSZ	1-2	F-000010*	CALP	1-2	6-000044
E	1-2	7-000010	EDF	1-2	F-000012*	HREAD	1-2	F-000024*
IEF	1-2	4-000050	IFRM	1-2	F-000004*	INS	1-2	F-000028*
LE	1-2	6-000006	LS	1-2	6-000004	LUN1	1-2	F-000032*
NC	1-2	6-000022	NCAR	1-2	6-000032	NSPR	1-2	6-000014
NDARS	1-2	6-000010	NS	1-2	6-000024	NSPR	1-2	6-000050
N	1-2	4-000004	RLB	1-2	4-000056	RSIZ	1-2	6-000036
SVD	1-2	6-000034						

## ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
BUFLER	1-1	F-000020*	000001	0 (1)
HEATER	1-1	F-000006*	000001	0 (1)
IFRM	1-2	4-000020	000014	6 (6)
IFR2	1-2	4-000034	000014	6 (6)
IFR3	1-2	4-000040	000004	2 (2)
LETRB	1-1	4-000000	000002	1 (2)
LILOC	1-2	4-000050	000005	3 (3)
MLD3	1-1	7-000005	000002	1 (2)
NAME	1-2	7-000000	000005	3 (3)
SCUD	1-1	4-000004	000014	6 (12)

## LABELS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LABL								
1		20	1-000250	30	1-000370	40	1-000540	45
50	1-001004	60	1-001134	70	1-001202	80	1-001230	90
100	1-001344	110	1-001362	9999				

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E

14:27:54

30-OCT-80

PAGE 6

HDREAD.FTN

/TR:BLOCKS/WR

FUNCTIONS AND SUBROUTINES REFERENCED

GETADR HDPCODE HPROS IBYTE WTGIO

TOTAL SPACE ALLOCATED = 001724 490

NO FPP INSTRUCTIONS GENERATED

,TT17:[320,40]A.B/-SP=SY0:[320,40]HDREAD

**ORIGINAL PAGE IS  
OF POOR QUALITY**

FORTRAN IV-PLUS V02-51E 14:27.25 30-OCT-80 PAGE 1  
HDCODE. FTN /TR: BLOCKS/WR

[illegible]

ORIGINAL PAGE IS  
OF POOR QUALITY

**14:27:25**

30-OCT-80

**PAGE 2**

HDCODE, FTN /TR: BLOCKS/WR

```
C      INPUTS:          FORMATS:
```

```
C      CCCCCCCC         CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

[illegible]

```
C      COMM          COMMON AREA FILLED WITH DATA ---  
C                               EXTRACTED FROM TAPE FOR USER
```

[illegible]

```
C ROUTINES CALLED:  
C  
C  
C ETAB, A2I, B2I, JULIAN, LATLON
```

C-\*\*\*\*\*

```

0001      SUBROUTINE HDCODE(IF,H,IE,LVL)
0002          REAL LATLON
0003          INTEGER DY,E,LATD,LONGD
0004          BYTE H(1)
0005          BYTE MON(3,12)
0006          BYTE HB
0007          INTEGER HI(32)
0008          REAL HR(16)
0009          INTEGER A21,B21
0010          INTEGER LVL
0011          INTEGER HOLD1
0012          INTEGER*4 HOLD2
0013          BYTE HLDB(4)
0014          BYTE HLDA(12),EXDAY(2),SCNID(12),MISNO(2)
0015          COMMON/COMH/HB(64)
0016          COMMON/LATLON/LATD,LONGD
0017          COMMON/LABEL/NAME(3),MISNO,E

```

C	HB IS ARRAY NAME FOR HEADER DATA ADDRESSED IN BYTES
C	HR IS ARRAY NAME FOR HEADER DATA ADDRESSED IN REAL FORM
C	HI IS ARRAY NAME FOR HEADER DATA ADDRESSED IN INTEGER FORM
C	HB, HR, AND HI ALL ADDRESS DATA IN ARRAY H
	EQUIVALENCE(HB(1),HI(1),HR(1))
	EQUIVALENCE(HLDA(1),HB(10))
	EQUIVALENCE(HLDB(1),HOLD2)
C	MON = ARRAY OF 3 LETTER EBCDIC CHARACTER ABBREVIATIONS
C	FOR MONTHS OF THE YEAR NAMES

0021 DATA MON/"321,"301,"325,"305,"305,"302,"324,"301,"331,  
1 "301,"327,"331,"324,"301,"350,"321,"344,"325,

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:27:25	30-OCT-80	PAGE 3
HDCODE. FTN		/TR: BLOCKS/WR		
		2 "321,"344,"323,"301,"344,"307,"342,"305,"327,		
		3 "326,"303,"343,"325,"326,"345,"304,"305,"303/		
0022	NAME(1)='HD'			
0023	NAME(2)='CO'			
0024	NAME(3)='DE'			
0025	IE = 0	!SET ERROR CODE NO ERROR		
0026	GO TO (1,2,3),IF			
C	SET ERROR CODE TO 1 AND RETURN TO MAIN PROGRAM			
0027	IE = 1			
0028	RETURN			
C	*****			
C	PROCESSING FOR LARSYS FORMAT TAPE			
C	*****			
0029	3 HI(1)=1	! ST (SCAN START) FIELD IN COMMON		
0030	HI(2)=B2I(H,23,24)	! SE (SCAN END) FIELD IN COMMON		
0031	HI(2)=HI(2)-6			
0032	HI(3)=1	! LS (LINE START)		
0033	HI(4)=2340			
0034	HI(5)=B2I(H,19,20)	! NOCHAN (# CHANNELS ON TAPE)		
0035	HI(6)=B2I(H,23,24)	! NOSAMP (# SAMPLES PER SCAN)		
0036	HI(6)=HI(6)-6			
C	BUILD SCNID (SCENE ID)			
0037	DO 31 I = 9,12			
0038	31 HLDB(I-8)=H(I)			
0039	HB(13)='040			
0040	DO 131 I=13,23			
0041	HB(I+1)=HB(I)			
0042	131 CONTINUE			
0043	ENCODE(12,300,HLDA)HOLD2			
0044	300 FORMAT(112)			
0045	CALL FRONT(HLDA,12)			
0046	HR(7)=0	!SUNEL AND SUNAZ		
C	SUNEL (SUN ELEVATION)	SUNAZ (SUN AZIMUTH)		
0047	HI(15)=B2I(H,63,64)	! SNSHD (SENSOR HEADING)		
0048	HI(16)=0	!NREV (ORBIT REVOLUTION NUMBER)		
0049	HI(17)=B2I(H,51,52)	!EXYR (EXPOSURE YEAR)		
0050	DY=B2I(H,47,48)			
0051	MO=B2I(H,43,44)			
0052	HI(18)=JULIAN(MO,DY,HI(17))	!EXDAY (DAY-JULIAN)		
0053	HI(19)=DY	!DAY (SCENE ID DAY)		
0054	HI(20)=A2I(H,53,54)	!HR (SCENE ID HOUR)		
0055	HI(21)=A2I(H,55,56)	!MIN (SCENE ID MIN)		
0056	HI(22)=0	!SEC (SCENE ID SECOND)		
0057	HI(23)=1	!SEGNO(1) (TAPE SEQUENCE #)		
0058	HI(24)=1	!SEGNO(2) (TAPE SEQUENCE #)		
0059	HR(13)=0	!FLAT (FORMAT CENTER LATITUDE)		
0060	HR(14)=0	!FLONG (FORMAT CENTER LONGITUDE)		
0061	HR(15)=0	!NLAT (NADIR LATITUDE)		
12 0062	HR(16)=0	!NLONG (NADIR LONGITUDE)		
11 0063	RETURN			
10 C	*****			
9 C	*****			
8 C	PROCESSING FOR LANDSAT FORMAT TAPE			
7 C	*****			
6 C	MISSION NUMER			

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14:27:25  30-OCT-80      PAGE 4
HDCODE PTN      /TR.BLOCKS/WR

C
0064      2 E=A2I(H,1,1)
0065      IF(E.EQ.1.OR.E.EQ.5)E=1
0066      IF(E.EQ.2.OR.E.EQ.6)E=2
0067      IF(E.NE.1.AND.E.NE.2)E=3
0068      N=H(14)
0069      HI(23)=IAND(N,15)      !SEGN0(1) (TAPE SEQUENCE #)
0070      WRITE(6,1311)HI(23)
0071      1311  FORMAT(1X,'TAPE SEQUENCE NO.=',I2)
0072      N=H(16)
0073      HI(24)=IAND(N,15)      !SEGN0(2) (TAPE SEQUENCE #)
0074      HI(6)=B2I(H,39,40)/4  !NOSAMP (# SAMPLES PER SCAN)
0075      HI(1)=HI(6)*(HI(23)-1)+1      !ST (SCAN START)
0076      HI(2)=HI(1)+HI(6)-1      !SE (SCAN END)
0077      HI(3)=1      !LS (LINE START)
0078      HI(4)=2340      !LE (LINE END)
0079      HI(5)=4      !NOCHAN (# CHANNELS ON TAPE)
0080      HI(17)=A2I(H,46,47)      !EXYR (EXPOSURE YEAR)
0081      DY=A2I(H,41,42)
C      *****
C      THIS ROUTINE CONVERTS EBCDIC MONTH NAMES TO
C      INTEGER MONTH NUMBER
C      *****
0082      DO 22 I = 1,12
0083      DO 25 J = 1,3
0084      IF(H(J+42).NE.MON(J,I)) GO TO 22
0085      25  CONTINUE
0086      GO TO 23
0087      22  CONTINUE
0088      H(18)=0      !EXDAY IF NO DATA IN HEADER
0089      GO TO 24
0090      23  HI(18)=JULIAN(I,DY,HI(17))      !EXDAY (DAY-JULIAN)
0091      CALL ETAB(H(51),LATD,1)
0092      24  HR(13)=LATLON(H,51,56)      !LAT (FORMAT CENTER LATITUDE)
0093      HR(14)=LATLON(H,58,64)      !FLONG (FORMAT CENTER LONGITUDE)
0094      CALL ETAB(H(58),LONGD,2)
C      TEST FOR CORRECT LANDSAT LEVEL
0095      IF ((H(135).AND."377").EQ."305")LVL=1      !LEVEL 1
0096      IF (LVL.EQ.1)GO TO 1000
0097      IF ((H(6).AND."377").EQ."140")LVL=3      !LEVEL 3
0098      IF (LVL.EQ.3)GO TO 3000
0099      LVL=2
C      PROCESS LEVEL 1
C      ETAB CONVERTS EBCDIC TO ASCII BYTE BY BYTE
C      BUILDS SCNID FIELD (SCENE ID)
0100      1000  CALL ETAB(H(1),HB(13),4)
0101      CALL ETAB(H(6),HB(17),2)
0102      CALL ETAB(H(11),HB(19),2)
0103      1100  DO 100 I=21,24
0104      HB(I)="040
0105      100  CONTINUE
0106      IF (LVL.NE.1)GO TO 2000
0107      HI(13)=A2I(H,101,102)      !SUNEL (SUN ELEVATION)
0108      HI(14)=A2I(H,106,108)      !SUNAZ (SUN AZIMUTH)
0109      HI(15)=A2I(H,110,112)      !SNSHD (SENSOR HEADING)
0110      HI(16)=A2I(H,114,117)      !NREV (ORBIT REVOLUTION #)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN I/-PLUS V02-51E      14:27:25      30-OCT-80      PAGE 5
HDCODE,FTN      /TR:BLOCKS/WR

0111      HI(19)=A2I(H,142,145)      !DAY (SCENE ID DAY)
0112      HI(20)=A2I(H,147,148)      !HR (SCENE ID HOUR)
0113      HI(21)=A2I(H,149,150)      !MIN (SCENE ID MINUTE)
0114      HI(22)=H(151)
0115      HI(22)=IAND(HI(22),15)*10      !SEC (SCENE ID SECOND)
0116      HR(15)=LATLON(H,68,73)      !NLAT (NADIR LATITUDE)
0117      HR(16)=LATLON(H,75,81)      !NLONG (NADIR LONGITUDE)
0118      RETURN
          C      PROCESS LEVEL 2
0119      2000      HI(13)=A2I(H,108,109)      !SUNEL
0120      HI(14)=A2I(H,112,114)      !SUNAZ
0121      HI(15)=A2I(H,67,69)      !SNSHD
0122      HI(16)=0      !NREV
0123      HI(19)=A2I(H,144,147)      !DAY
0124      HI(20)=A2I(H,149,150)      !HR
0125      HI(21)=A2I(H,151,152)      !MIN
0126      HI(22)=H(153)
0127      HI(22)=IAND(HI(22),15)*10      !SEC
0128      HR(15)=LATLON(H,77,82)      !NLAT
0129      HR(16)=LATLON(H,84,90)      !NLONG
0130      RETURN
          C      PROCESS LEVEL 3
          C      BUILD SCNID
0131      3000      CALL ETAB(H(1),HB(13),5)
0132      CALL ETAB(H(6),HB(18),2)
0133      CALL ETAB(H(12),HB(20),1)
0134      GO TO 1100
          C      *****
          C      PROCESSING FOR UNIVERSAL FORMAT TAPE
          C      *****
0135      1      HI(1)=1      !ST (START SCAN)
0136      HI(3)=0      !LS (LINE START)
0137      HI(4)=0      !LE (LINE END)
0138      HI(5)=H(90)      !NOCHAN (# CHANNELS ON TAPE)
0139      HI(6)=B2I(H,1787,1788)      !NOSAMP (# SAMPLES PER SCAN)
0140      HI(2)=HI(6)      !SE (SCAN END)
0141      MISNO(1)=H(65)
0142      MISNO(2)=H(66)
0143      HOLD1=B2I(H,67,68)      !SCNID (SCENE ID)
0144      DO 111 I=13,24
0145      HB(I)='040
0146      111      CONTINUE
0147      ENCODE(6,200,HLDA)HOLD1
0148      200      FORMAT(115)
0149      CALL FRONT(HLDA,12)
0150      ENCODE(6,200,SCNID)HOLD1
0151      HI(13)=A2I(H,2163,2164)      !SUNEL (SUN ELEVATION)
0152      HI(14)=A2I(H,2159,2170)      !SUNAZ (SUN AZIMUTH)
12 0153      HI(15)=0      !SNSHD (SENSOR HEADING)
1 0154      HI(16)=B2I(H,71,72)      !NREV (ORBIT REVOLUTION #)
0 0155      HI(17)=H(80)      !EXYR (EXPOSURE YEAR)
0 0156      DY=H(78)
0 0157      ENCODE(2,250,EXDAY)DY
7 0158      250      FORMAT(12)
0159      SCNID(7)=EXDAY(1)
0160      SCNID(8)=EXDAY(2)

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14:27:25      30-OCT-80      PAGE 6
HDCODE. FTH      /TR: BLOCKS/MR

0161      SCNID(9)=' '
0162      SCNID(10)='1'
0163      SCNID(11)='H'
0164      SCNID(12)='C'
0165      CALL FRGNT (SCNID,12)
0166      MO=H(79)
0167      HI(18)=JULIAN(MO,DY,HI(17))      !EXDAY (DAY-JULIAN)
0168      HI(19)=A2I(H,2124,2126)      !DAY (SCENE ID DAY)
0169      HI(20)=A2I(H,2127,2128)      !HR (SCENE ID HOUR)
0170      HI(21)=A2I(H,2129,2130)      !MIN (SCENE ID MINUTE)
0171      HI(22)=H(2131)
0172      HI(22)=IAND(HI(22),15)*10      !SEC (SCENE ID SECOND)
0173      HI(23)=1      !SEQND(1) (TAPE SEQUENCE #)
0174      HI(24)=1      !SEQND(2) (TAPE SEQUENCE #)
0175      HR(13)=LATLON(H,2134,2139)      !FLAT (FORMAT CENTER LATITUDE)
0176      HR(14)=LATLON(H,2140,2145)      !FLONG (FORMAT CENTER LONGITUDE)
0177      HR(15)=0      !NLAT (NADIR LATITUDE)
0178      HR(16)=0      !NLONG (NADIR LONGITUDE)
0179      RETURN
0180      END

```



PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	004314	1126 RW, I, CON, LCL
2	\$PDATA	000454	150 RW, D, CON, LCL
3	\$IDATA	000712	229 RW, D, CON, LCL
4	\$VARS	000102	33 RW, D, CON, LCL
5	\$TERPS	000002	1 RW, D, CON, LCL
6	\$CPTH	000100	32 RW, D, OVR, GBL
7	\$LATLON	000004	2 RW, D, OVR, GBL
8	\$LABEL	000012	5 RW, D, OVR, GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
HCODE		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
IV	I*2	4-000004	E	I*2	8-000010	HOLD1	I*2	4-000052
IE	I*2	F-000006*	IF	I*2	F-000002*	J	I*2	4-000100
LVL	I*2	F-000010*	MD	I*2	4-000074	N	I*2	4-000076

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
E/DAY	L*1	4-000054	000002	1 (2)
H	L*1	F-000004*	000001	0 (1)
HE	L*1	6-000030	000100	32 (64)
HI	I*2	6-000000	000100	32 (32)
HLDA	L*1	6-000014	000014	6 (12)
HLDB	L*1	4-000000	000004	2 (4)
HR	R*4	6-000000	000100	32 (16)
HLGND	L*1	8-000006	000002	1 (2)
MDN	L*1	4-000006	000044	18 (3,3)
NAME	I*2	8-000000	000006	3 (3)
SCNID	L*1	4-000056	000014	6 (12)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-003236	2	1-000732	3	1-000112	22	1-001460
24	1-001565	25	**	31	**	100	**
131	**	200	3-000034	250	3-000040	300	3-000000
1100	1-002132	1311	3-000004	2000	1-002560	3000	1-003120

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E

14:27:25

30-OCT-80

PAGE 8

HDCODE.FTN

/TR:BLOCKS/WR

FUNCTIONS AND SUBROUTINES REFERENCED

A2I B2I ETAB FRONT JULIAN LATLON

TOTAL SPACE ALLOCATED = 006124 1578

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:27:46 30-OCT-80 PAGE 9

```
C *****
C *****
C THIS ROUTINE CONVERTS 2 BYTE DATA FIELD FROM
C   BINARY TO INTEGER
C *****
```

```
0001      INTEGER FUNCTION B2I(H,B1,B2)
0002      BYTE H(1)
0003      INTEGER B1,B2,C,D
0004      C=H(B1)
0005      D=H(B2)
0006      B2I=IAND(C,255)*256+IAND(D,255)
0007      RETURN
0008      END
```

ORIGINAL PAGE 10  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:27:46 30-OCT-60 PAGE 10  
MODULE FVN /TR: BLOCKS/MR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000112	37 RM.I. CON. LCL
3	\$IDATA	000014	6 RM.D. CON. LCL
4	\$VARS	000004	2 RM.D. CON. LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
B21	I*2	1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
B1	I*2	F-000004*	B2	I*2	F-000006*	C	I*2	4-000000
						D	I*2	4-000002

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
H	L*1	F-000002*	000001	0 (1)

TOTAL SPACE ALLOCATED = 000132 45

NO FPP INSTRUCTIONS GENERATED

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-91E

14:27:48

30-OCT-80

PAGE 11

HDCODE. FTY

/TR: BLOCKS/WR

C \*\*\*\*\*

C THIS ROUTINE CONVERTS DATA FROM BCD TO INTEGER

C \*\*\*\*\*

0001 INTEGER FUNCTION A2I(H,B1,B2)

0002 BYTE H(1)

0003 INTEGER B1,B2,C

0004 A2I=0

0005 DO 1 I = B1,B2

0006 C=H(I)

0007 C=IAND(C,15)

0008 IF(C.GT.9)C=0

0009 1 A2I=A2I\*10+C

0010 RETURN

0011 END

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000152	53 RM. I. CON. LCL
3	\$IDATA	000014	6 RM. D. CON. LCL
4	\$VARS	000004	2 RM. D. CON. LCL
5	\$TEMPS	000002	1 RM. D. CON. LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
A21	I*2	1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
D1	I*2	F-000004*	B2	I*2	F-000006*	C	I*2	4-000000

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
H	L*1	F-000002*	000001	C (1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1					

TOTAL SPACE ALLOCATED = 000174 62

NO FPP INSTRUCTIONS GENERATED

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTHAN IV-PLUS V02-51E		14.27.49	30-OCT-80	PAGE 13
HDCODE. FTH		/TR: BLOCKS/WR		
	C *****			
	C *****			
	C THIS ROUTINE COMPUTES LATITUDE AND LONGITUDE IN DEGREES			
	C *****			
0001	REAL FUNCTION LATLON(H, B1, B2)			
0002	IMPLICIT INTEGER (A-Z)			
0003	BYTE HB			
0004	BYTE H(1)			
0005	BYTE N, S, E, W, D			
0006	DATA N, S, E, W, D/213, 226, 197, 230, 96/			
0007	LL = 0			
0008	SIGN = 0			
0009	DO 1 I = B1, B2			
0010	HB=H(I)			
0011	IF(HB, EQ, N) GO TO 1			
0012	IF(HB, EQ, S) GO TO 3			
0013	IF(HB, EQ, E) GO TO 1			
0014	IF(HB, EQ, W) GO TO 3			
0015	IF(HB, NE, D) GO TO 4			
0016	GO TO 1			
0017	3 SIGN = 1			
0018	GO TO 1			
0019	4 J = HB			
0020	J = IAND(J, 15)			
0021	LL = LL*10+J			
0022	1 CONTINUE			
0023	LM = LL - (LL / 100) * 100			!MINUTES CALCULATION
0024	LL = LL / 100			!DEGREES CALCULATION
0025	LATLON = FLOAT( LM ) / 60. + FLOAT( LL )			
	C LATLON CONVERTS DATA TO REAL VARIABLE FOR RETURN			
	C SIGN IS NEGATIVE FOR S OR W DIRECTION			
0026	IF( SIGN, NE, 0 ) LATLON = - LATLON			
0027	RETURN			
0028	END			

ORIGINAL PAGE  
OF POOR QUALITY

7TR BLOCKS/HR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1	000370	RM. I. CON. LCL
2	\$IDATA	000016	RM. D. CON. LCL
4	\$VARS	000020	RM. D. CON. LCL
5	\$TEMP5	000002	RM. D. CON. LCL

ENTER POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LATLON R#4		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
B1	1*2	F-000004*	B2	1*2	F-000004*	D	1*1	4-000003
J	1*2	4-000012	J	1*2	4-000014	LL	1*2	4-000016
S	1*1	4-000002	SIGN	1*2	4-000010	W	1*1	4-000004

APPENDIX

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
H	1*1	F-000002*	000001	0 (1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000242	3	1-000160	4	1-000176		

TOTAL SPACE ALLOCATED = 000430 140

1117 (320.40)A B7-SP-SV0 (320.40)HRCODE



ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14 26 41 30-OCT-80 PAGE 1  
LOCODE FTN /TR BLOCKS/WA

LOCKHEED ELECTRONICS COMPANY, INC.  
REMOTE SENSING APPLICATIONS LAB.  
( R S A L )

NAME: LDCODE

DEVELOPED BY: SUSANNE OBRIEN  
ON: 6/78

REVISED BY: ON: PURPOSE FOR REVISION:

**PROGRAM DESCRIPTION:**

THIS ROUTINE WILL DECODE VIDEO DATA FROM A FOREIGN IMAGERY  
TAPE IN ONE OF THE FOLLOWING FORMATS: UNIVERSAL, LANDSAT 1 OR 2,  
OR LARSYS 2 OR 3  
ENTRY POINTS:

PROGRAM INITIATION OR CALL SEQUENCE:

```
CALL LDCODE(LUN, IF, BUFFER, BFSZ, L, RCHAN, LSS, LSE, BUF, EOF,  
1 PC, IE, INIT)
```

**CALL ARGUMENTS:**

**FORMATS:**

**LUN  
IF**

11 LOGICAL UNIT NUMBER  
IF FORMAT OF THE INPUT TAPE

1=UNIVERSAL  
2=LANDSAT  
3=LARSYS

**BUFFER**

BYTE STRING BUFFER FOR INTERNAL USE  
OF LDCODE

**BFSZ**

```

11 SIZE OF BUF IN BYTES
11 REQUESTED LINE NUMBER

```

**RCHAN**

11 REQUESTED CHANNEL NUMBER

**LSS**

# I1 FIRST PIXEL TO MOV

LSE  
PUE

```

11 LAST PIXEL IN MOVE
BYTE STRING BUFFERS FOR DECODED PIXELS

```

BOF  
E0F

END OF FILE LEAF

PC

11 PARITY COUNT

**IE**

## II ERROR CODE

INI

11 FIRST PASS SWITCH 0=FIRST PASS

INPUTS:  
0000

## FORMATS

HC DM

COMMON BLOCK OF DATA FROM HBBOS

OUT

COMMON BLOCK OF DATA FROM AFROS  
FORMATS:

**Abstract**

\_\_\_\_\_

**ORIGINAL PAGE IS  
OF POOR QUALITY**

[illegible]

ORIGINAL PAGE  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:26:41 30-OCT-80 PAGE 3  
LDCOD...FTN /TR:BLOCKS/WR

```

0029      IF (IF EQ 2) GO TO 1      'FORMAT =LANDSAT
          C      THIS ROUTINE BUILDS ARRAYS IB AND JR
          C      IB(I)=OFFSET BYTES TO CHAN I
          C      JR(I)=RECORD # WHERE CHAN I IS FOUND
0030      IF (NRPC GT 1) GO TO 100      !#RECS/CHAN >1
0031      IF (NCAR GT 0) GO TO 10      !DATA IN ANCL REC GO
          C      THIS ROUTINE FOR NO DATA IN ANCL REC FIRST CHAN IN REC 2
0032      JR(1)=2
0033      IB(1)=2
0034      DO 140 I=2,NC
0035      JR(I)=JR(I-1)+1
0036      IB(I)=2
0037      140      CONTINUE
0038      GO TO 1
0039      10      FC = 1
0040      LC = NCAR
0041      I = 1
0042      IR = 1
0043      DO 12 CHAN = 1,NC
0044      13      CONTINUE
0045      IF (IR GT 1) ANC = 2+SVD
0046      IF (CHAN GE FC AND CHAN LE LC) GO TO 14
0047      IF (CHAN LE LC OR IR GE NRPDS) GO TO 15
          C      THIS ROUTINE RESETS VARIABLES FOR NEW RECORD NEEDED
0048      FC = LC + 1
0049      LC = LC + NCPR
0050      IR = IR + 1
0051      GO TO 13
          C      THIS ROUTINE CALCULATES OFFSET BYTES AND RECORD #
0052      14      IB(I)=(CHAN -FC)*(NS+CALP)+ANC-1
0053      JR(I)=IR
0054      16      I = I + 1
0055      12      CONTINUE
0056      GO TO 1
0057      15      CONTINUE
0058      CERR = 4      !ERROR TOO MANY CHANS OR RECS
0059      GO TO 50
          C      THIS ROUTINE SETS UP ARRAYS FOR MULTIPLE RECS/CHAN
0060      100      IF (NCAR GT 0) GO TO 120
0061      JR(1)=2
0062      IB(1)=2
0063      110      DO 130 I=2,NC
0064      JR(I)=JR(I-1)+NRPC
0065      IB(I)=2
0066      130      CONTINUE
0067      GO TO 1
0068      120      JR(1)=1
0069      IB(1)=ANCL+SVD-1
2 0070      GO TO 110
1 0071      C      START IF NOT FIRST PASS OR IS LANDSAT
0 0072      1      IF (RCHAN LE NC) GO TO 31 !REQ CHAN TOO BIG?
9 0073      30      CONTINUE
8 0074      CERR=6 !YES ERROR
7 0075      GO TO 50
6 0076      31      CONTINUE
          IF (CL-DLIN) 29,17,18      !CURRENT LINE=DESIRED LINE

```

ORIGINAL COPY IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:24:41	30-OCT-80	PAGE 4
LDCODE.FTN		/TR.BLOCKS/WR		
0077	18	CONTINUE		
0078		WRITE(6,1111)CL,DLIN		
0079	1111	FORMAT(1X,'CL=',15,'DLIN=',15)		
0080		CERR=7 !NO ERROR CL>DL		
0081		GO TO 50		
	C	ROUTINE FOR NEED NEXT LINE CL<DL		
	C	IF # DATA SETS PER RECORD < OR =1, NEED NEXT RECORD		
0082	29	IF (NDSPR.LE.1)GO TO 43		
	C	ROUTINE FOR # DATA SETS PER RECORD >1		
	C	CALC FIRST LINE OF DESIRED REC		
0083		FLIN=DLIN-MOD(DLIN-ONEST,NDSPR)		
0084		ADD=(DLIN-FLIN)*DSL		
0085		LIM=(FLIN-CL)/NDSPR		
0086		IF(MOD((FLIN-CL),NDSPR).NE.0)LIM=LIM+1		
0087		IF(LIM.EQ.0)GO TO 190		
0088		GO TO 44		
	C			
0089	43	CONTINUE		
0090		LIM=(DLIN-CL-1)*NRPDS+1		
	C			
	C	READING DATA FROM TAPE READ LIM RECORDS		
	C			
0091	44	DO 119 I=1,LIM		
0092		CALL WTQIO(RLB,LUN1,1,,ISTAT,IPRM,ISW)		
	C			
0093		IF(ISTATB(1).EQ."001")GO TO 119 !READ OKAY=1		
0094		IF(ISTATB(1).EQ.IEEOF)EOF=1 !END OF FILE="366"--10		
0095		IF(EOF.EQ.1)RETURN		
0096		IF(ISTATB(1).EQ.IEPC)PC=PC+1 !PARITY CHECK="374"--4		
0097		IF(ISTATB(1).EQ.IEPC)GOTO 119		
0098		GOTO 60 !READ ERROR		
0099	119	CONTINUE		
	C			
	C	SET CL TO DESIRED LINE AND CCAN TO 1		
	C			
0100	190	CL=DLIN		
0101		CCAN = 1		
	C	THIS ROUTINE FOR CURRENT LINE =DESIRED LINE CL=DL		
0102	17	IF(NRPDS.LE.1) GO TO 23 !#RECS/DATA SET <=1		
	C	IF NO THEN DETERMINE IF REG REC IN BUFFER		
	C	THIS CHECK FOR NO DATA IN ANCIL REC		
0103		IF(RCHAN.NE.1)GO TO 170 !REG CHAN NOT =1 GO		
0104		IF((LSS.EQ.1).AND.(NCAR.EQ.0)) GO TO 180		
0105	170	I = JR(RCHAN)-JR(CCAN)		
0106		CCAN = RCHAN		
0107		IF(I.EQ.0) GO TO 23 !REG REC IN BUFFER		
	C			
0108	180	CONTINUE		
12	C			
11	C	READING DATA FROM TAPE		
10	C			
9	0109	CALL WTQIO(RLB,LUN1,1,,ISTAT,IPRM,ISW)		
8	C			
7	0110	IF(ISTATB(1).EQ."001")GO TO 23		
6	0111	IF(ISTATB(1).EQ.IEEOF)EOF=1		
5	0112	IF(EOF.EQ.1)RETURN		
4				

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-51E      14:26:41    30-OCT-80      PAGE 5
LD CODE,FTN      /TR:BLOCKS/WR

0113      IF(ISTATB(2).EQ.IEPC)PC=PC+1
0114      IF(ISTATB(1).EQ.IEPC)GO TO 23
0115      GOTO 60      !READ ERROR
0116      23      IF(IF.EQ.2) GO TO 25      !LANDSAT?
C      THIS ROUTINE MOVES DATA FOR UNIVERSAL/LARSYS FORMAT
0117      IF(NRPC.GT.1)GO TO 300      !MULTIPLE RECS/CHAN
0118      B = IB(RCHAN)+LSE      +ADD
0119      E = B+LSE-LSS
0120      K = 0
0121      DO 26 I = B,E
0122      K = K + 1
0123      26      BUF(K)=BUFFER(I)
0124      GO TO 50
C      THIS ROUTINE MOVES DATA FOR LANDSAT FORMAT
0125      25      B=((LSS-1)/2)*8+RCHAN*2-MOD(LSS,2)
0126      E=((LSE-1)/2)*8+RCHAN*2-MOD(LSE,2)
0127      J = 1+MOD(LSS,2)
0128      K = 0
0129      28      DO 27 I = 1,J
0130      K = K + 1
0131      BUF(K)=BUFFER(B)
0132      B = B + 1
0133      27      CONTINUE
0134      IF(B.GT.E)GO TO 50
0135      J = 2
0136      B = B + 6
0137      GO TO 28
C      THIS ROUTINE FOR MULTIPLE RECS/CHAN
0138      200      IF(LSS.NE.1)GO TO 210
0139      Z=NSPR      !Z=CONTROL VARIABLE FOR ROUTINE
0140      Y=LSE      !Y=CONTROL VARIABLE FOR UPPER LIMIT
0141      IF(LSE.GT.NSPR)Y=NSPR
0142      B=IB(RCHAN)+LSS
0143      E=B+Y-1
0144      K=0
0145      230      DO 220 I=B,E
0146      K=K+1
0147      BUF(K)=BUFFER(I)
0148      220      CONTINUE
0149      IF(LSE.LE.2)GO TO 50
C
C      READ DATA FROM TAPE
C
0150      CALL WTGID(RLB,LUN1,1,ISTAT,IPRM,ISW)
0151      IF(ISTATB(1).EQ."001")GO TO 223
0152      IF(ISTATB(1).EQ.IEOF)EOF=1
0153      IF(EOF.EQ.1) RETURN
0154      IF(ISTATB(1).EQ.IEPC)PC=PC+1
12 0155      IF(ISTATB(1).EQ.IEPC)GO TO 223
11 0156      GOTO 60      !READ ERROR
10 0157      223      Z=Z+NSPR
9 0158      IF(LSE.LT.2)Y=LSE-(Z-NSPR)
9 0159      B=3
7 0160      E=2+Y
0161      GO TO 230
C      THIS ROUTINE FOR SECOND PASS PROC FOR NRPC>1

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTTRAN IV-PLUS V02-51E 14:26:41 30-OCT-80 PAGE 6  
LDCODE.FTN /TR:BLOCKS/WA

```
0162 210 Z=NSPR*(LSS/NSPR+1)
0163      B=2+LSS-(Z-NSPR)
0164      Y=NSPR
0165      IF(LSE.LT.Z)Y=LSE-(Z-NSPR)
0166      E=2+Y
0167      K=0
0168      GO TO 230
0169 50 IE = CERR !SET ERROR FOR RETURN
0170      RETURN
0171 60 IE=ISTATB(1) !SET READ ERROR
0172      RETURN
0173      END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	SCODE1	003406	899
2	SPDATA	000004	2
3	SDATA	000074	30
4	SVARS	000506	163
5	SIEMPS	000010	4
6	SIEM	000052	21
7	LABEL	000012	5

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LCODE	1-000000							

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ACD	1-2	4-000432	AMC	1-2	4-000452	ANCL	1-2	6-000020
CALP	1-2	5-000004	CCAN	1-2	4-000450	CEFR	1-2	6-000046
DLIN	1-2	4-000012	DOI	1-2	6-000030	DSL	1-2	4-000440
FC	1-2	4-000456	FLIN	1-2	4-000466	HSIZ	1-2	6-000042
TEGUF	1-2	4-000432	IIPC	1-2	4-000434	IF	1-2	6-000042
ISH	1-2	4-000472	J	1-2	4-000500	K	1-2	4-000476
LIN	1-2	4-000470	LS	1-2	6-000004	LSE	1-2	6-000020
MBIT	1-2	6-000036	NC	1-2	6-000010	NCAR	1-2	6-000024
MRPC	1-2	6-000016	MRPDS	1-2	6-000010	NS	1-2	6-000024
PC	1-2	6-000026	PRSZ	1-2	4-000436	PSKIP	1-2	6-000040
RSIZ	1-2	6-000036	SCNID	1-2	4-000012	SE	1-2	6-000002
Y	1-2	4-000504	Z	1-2	4-000502			

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
EUF	1-1	F-000022	000001	0 (1)
IB	1-2	F-000006	000001	0 (1)
IPM	1-2	4-000014	000200	64 (64)
ISTAT	1-2	4-000014	000014	6 (6)
ISTATB	1-1	4-000030	000004	2 (2)
JR	1-2	4-000014	000200	64 (64)
LSLOC	1-2	4-000034	000006	3 (3)
MTSD	1-1	7-000006	000002	1 (2)
NAFL	1-2	7-000006	000006	3 (3)

LEADS

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL COPY  
OF PROGRAM QUALITY

1-1000000000 PLUS 002-51E 14 26 41 30-OCT-80 PAGE 8  
LARGE FPN /TR. BLOCKS/MR

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-001132	9	**	10	1-000450	12	**	13	1-000322
14	1-000742	15	1-000742	16	**	17	1-001644	18	**
23	1-001126	25	1-002304	26	**	27	**	28	1-002450
29	1-001160	30	**	31	1-001162	43	1-001414	44	1-001450
50	1-001350	60	1-003366	100	1-000762	110	1-001612	119	1-001650
120	1-001076	130	**	140	**	170	1-001730	180	1-002000
190	1-001242	200	1-002574	210	1-003216	220	**	223	1-003124
230	1-002710	1111	3-000000						

FUNCTIONS AND SUBROUTINES REFERENCED

DETAILS WTCIO

TOTAL SPACE ALLOCATED = 004310 1124

H3 FFP INSTRUCTIONS GENERATED

1117 (320,401A B/-SP-SYO (320,401DCODE



## APPENDIX D

### LISTING OF POSTP PROCESSOR

ORIGINAL PAGE  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:33:33	30-OCT-80	PAGE 1
POSTPP.FTN		/TR:BLOCKS/WR		
0001		PROGRAM POSTPP		
	C			
	C	THIS PROGRAM IS THE POST PATCH PROCESSING PROCESSOR		
	C	IT COMPUTES THE SPATIAL RESPONSE FUNCTION BY REGRESSION.		
	C	ONTO A QUADRATIC SURFACE		
	C			
0002		BYTE FILNAM(30),REPORT(4),ID(10),NAME(6)		
0003		DIMENSION IGAR(16,16),ICLOUD(16,16),IWATER(16,16)		
0004		DIMENSION VIN(5,16,16),IGIN(3,16,16),VINH(16,16,2),R(16,16)		
0005		DIMENSION P(6),C(6),X(4)		
	C			
	C	READ FILE NAME FROM INPUT CARD		
	C			
0006		READ(5,1000)FILNAM		
0007	1000	FORMAT (30A1)		
0008		DO 10 I=1,30		
0009		IF(FILNAM(I).NE.' ')GO TO 10		
0010		FILNAM(I)=0		
0011		GO TO 15		
0012	10	CONTINUE		
	C			
	C	READ REPORT TYPE REQUEST FROM CARD SHOULD BE LONG OR SHORT		
	C			
0013	15	READ(5,1200)REPORT		
0014	1200	FORMAT(4A1)		
0015		IPASS=1		
	C			
	C	OPEN FILE AND READ HEADER AND DATA		
	C			
0016		OPEN(UNIT=1,NAME=FILNAM,TYPE='OLD',FORM='FORMATTED')		
0017		READ(1,2000)ID,IAQ,ISAT,LATDR,LATD,LATM,LONDR,LOND,LONM,NAME		
0018	2000	FORMAT(1X,10A1,I2,I1,A1,I4,I2,A1,I4,I2,2X,6A1)		
0019		DO 25 I=1,16		
0020		DO 25 J=1,16		
0021		READ(1,1100)IPATCH,(X(K),K=1,4),IGAR(I,J),ICLOUD(I,J), 1 IWATER(I,J),GAMA,ISLN,(VIN(K,I,J),K=1,5),(IGIN(K,I,J),K=1,3)		
0022	1100	FORMAT(1X,I3,4F6,2,3I4,F6,3,I3,3F4,1,2F5,3,3I3)		
0023	25	CONTINUE		
	C			
	C	SET UP RN** CONSTANTS REGRESSION CONSTANTS		
	C			
0024		RN00=16		
0025		RN01=5440.**.5		
0026		RN02=91392.**.5		
0027		RN11=115600.**.5		
	C			
	C	CALCULATE C VALUES		
	C			
12	0028	30	DO 40 I=1,6	
11	0029		C(I)=0	
10	0030	40	CONTINUE	
9	0031		IGOOD=0	
8	0032		DO 100 I=1,16	
7	0033		DO 100 J=1,16	
6	0034		IF(IPASS.EQ.2)GO TO 42	
5		C	ISUM=COUNT OF BAD PIXELS IN PATCH	

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14. 33. 33	30-OCT-80	PAGE 2
POSTPP. FTN		/TR BLOCKS FOR		
0035		ISUM=IGAR(I,J)+ICLOUD(I,J)+IWATER(I,J)		
0036		IF(ISUM GT 128)GO TO 100		
0037	42	IGOOD=IGOOD+1		
0038		IF(IPASS.EQ.1)VAR=IGIN(I,I,J)		
0039		IF(IPASS.EQ.2)VAR=VINH(I,J,1)		
	C			
	C	P(K) ORTHOGONAL POLYNOMIALS		
	C			
0040		P(1)=1/RN00		
0041		P(2)=1/RN01*(I-8.5)		
0042		P(3)=1/RN01*(J-8.5)		
0043		P(4)=1/RN02*((I-8.5)**2-21.25)		
0044		P(5)=1/RN02*((J-8.5)**2-21.25)		
0045		P(6)=1/RN11*(I-8.5)*(J-8.5)		
	C			
	C	C(K) RAW ORTHOGONAL COEFFICIENTS		
	C			
0046		DO 50 K=1,6		
0047		C(K)=C(K)+P(K)*VAR		
0048	50	CONTINUE		
0049	100	CONTINUE		
0050		DO 150 K=1,6		
0051		C(K)=C(K)*256./IGOOD		
0052	150	CONTINUE		
	C			
	C	CALCULATE VINHAT, R, SSR, AND MSE		
	C	VINH VINHAT IS THE ESTIMATED VARIABLE VALUE		
	C	SSR IS THE SUM OF THE SQUARES OF THE RESIDUALS		
	C	RMSE IS THE ROOT MEAN SQUARE ERROR		
	C			
0053		SUM=0		
0054		SSR=0		
0055		DO 200 I=1,16		
0056		DO 200 J=1,16		
0057		IF(IPASS.EQ.1)VAR=IGIN(I,I,J)		
0058		IF(IPASS.EQ.2)VAR=VINH(I,J,1)		
0059		P(1)=1/RN00		
0060		P(2)=1/RN01*(I-8.5)		
0061		P(3)=1/RN01*(J-8.5)		
0062		P(4)=1/RN02*((I-8.5)**2-21.25)		
0063		P(5)=1/RN02*((J-8.5)**2-21.25)		
0064		P(6)=1/RN11*(I-8.5)*(J-8.5)		
0065		DO 175 K=1,6		
0066		VINH(I,J,IPASS)=VINP(I,J,IPASS)+C(K)*P(K)		
0067	175	CONTINUE		
0068		R(I,J)=VAR-VINH(I,J,IPASS)		
0069		IF(IPASS.EQ.2)GO TO 178		
0070		ISUM=IGAR(I,J)+ICLOUD(I,J)+IWATER(I,J)		
0071		IF(ISUM GT 128)GO TO 200		
0072	178	SSR=SSR+R(I,J)**2		
0073		SUM=SUM+R(I,J)		
0074	200	CONTINUE		
0075		RMSE=SSR/(IGOOD-6)		
0076		RMSE=RMSE** .5		
	C			
	C	CALCULATE CN** VALUES COEFFICIENTS FOR THE ORTHOGONAL POLYNOMIALS		

ORIGINAL PAGE IS  
OF POOR QUALITY

```

FORTRAN IV-PLUS V02-91E      14:33:33      30-OCT-80      PAGE 3
POSTPP.FTN      /TR-BLOCKS/WR

      C
0077      CN00=C(1)/RN00
0078      CN01=C(2)/RN01
0079      CN10=C(3)/RN01
0080      CN02=C(4)/RN02
0081      CN20=C(5)/RN02
0082      CN11=C(6)/RN11

      C
      C      CALCULATE SAS VALUES
      C      COEFFICIENTS FOR SIMPLE POLYNOMIALS TO BE INPUT TO SAS
      C
0083      SAS1=CN00-8.5*(CN01+CN10)+51.*(CN02+CN20)+72.25*CN11
0084      SASX=-CN01+17.5*CN02+8.5*CN11
0085      SASY=CN10-17.5*CN20-8.5*CN11
0086      SASX2=CN02
0087      SASY2=CN20
0088      SASXY=-CN11

      C
      C      WRITE SHORT REPORT INFO
      C
      C
0089      IF(IPASS.EQ.2)GO TO 600
0090      IPASS=IPASS+1
0091      WRITE(6,1300)ID,NAME
0092      1300      FORMAT(1H1,40X,'PATCH REPORT FOR SCENE',1X,10A1,2X,'FILE:',6A1,7/)
0093      WRITE(6,1400)IAG,ISAT,LATDR,LATD,LATM,LONDR,LOND,LONM
0094      1400      FORMAT(5X,'SUN ANGLE:',15,5X,'SATELLITE NUMBER:',15,
1 3X,'LATITUDE:',1X,A1,1X,I4,'DEGREES',1X,I2,'MINUTES',
1 5X,'LONGITUDE:',1X,A1,1X,I4,'DEGREES',1X,I2,'MINUTES',/)
0095      WRITE(6,1500)(C(K),K=1,6),RMSE
0096      1500      FORMAT(1X,'C00=',F8.2,3X,'C01=',F8.2,3X,'C10=',F8.2,3X,
1 'C02=',F8.2,3X,'C20=',F8.2,3X,'C11=',F8.2,3X,
2 'MSE=',F8.2)
0097      WRITE(6,1550)CN00,CN01,CN10,CN02,CN20,CN11,SUM
0098      1550      FORMAT(1X,'CN00',F8.4,3X,'CN01',F8.4,3X,'CN10',
1 F8.4,3X,'CN02',F8.4,3X,'CN20',F8.4,3X,'CN11',
1 F8.4,3X,'SUMSR',F8.2)
0099      WRITE(6,1575)SAS1,SASX,SASY,SASX2,SASY2,SASXY
0100      1575      FORMAT(1X,'SAS1',F8.4,3X,'SASX',F8.4,3X,'SASY',
1 F8.4,3X,'SASX2',F8.4,3X,'SASY2',F8.4,3X,'SASXY',
2 F8.4)
0101      IF(REPORT(1).EQ.'S')GO TO 500

      C
      C      WRITE LONG REPORT DATA
      C
0102      WRITE(6,1600)
0103      1600      FORMAT(/,10X,'PATCH',10X,'VIN',10X,'VINHAT',12X,'R')
0104      DO 400 I=1,16
0105      DO 400 J=1,16
0106      IP=(I-1)*16+J
0107      WRITE(6,1700)IP,IGIN(1,I,J),VINH(I,J,1),R(I,J)
0108      1700      FORMAT(11X,I3,11X,I3,8X,F8.2,8X,F8.2)
0109      400      CONTINUE
0110      500      CONTINUE

      C
      C      USE THIS GO TO FOR 2 PASS RESULTS FOR CHECK OUT OF ALGORITHM

```

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E		14:33:33	30-OCT-80	PAGE 4
POSTPP.FTN		/TR:BLOCKS/WR		
0111	C	GO TO 30		
	C	GO TO 700		
	C	WRITE REPORT FOR PASS 2		
0112	600	WRITE(6,1800)		
0113	1800	FORMAT(1H1)		
0114		WRITE(6,1500)(C(K),K=1,6),RMSE		
0115		WRITE(6,1550)CNO0,CNO1,CN10,CNO2,CN20,CN11,SUM		
0116		IF(REPORT(1).EQ.'S')GO TO 700		
0117		WRITE(6,1600)		
0118		DO 650 I=1,16		
0119		DO 650 J=1,16		
0120		IP=(I-1)*16+J		
0121		WRITE(6,1750)IP,VINH(I,J,1),VINH(I,J,2),R(I,J)		
0122	1750	FORMAT(11X,13,6X,F8.2,8X,F8.2,8X,F8.2)		
0123	650	CONTINUE		
0124	700	CONTINUE		
0125		END		

ORIGINAL PAGE IS  
OF POOR QUALITY

1 FORTRAN IV-PLUS V02-51E 14:33:33 30-OCT-80 PAGE 5  
POSTP:FIN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	\$CODE1 004506	1187	RW, I, CON, LCL
2	\$PDATA 000014	6	RW, D, CON, LCL
3	\$IDATA 001120	296	RW, D, CON, LCL
4	\$VARS 026350	5748	RW, D, CON, LCL
5	\$TEMPS 000002	1	RW, D, CON, LCL

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CN02	R*4	4-026266	CN01	R*4	4-026272	CN02	R*4	4-026302	CN10	R*4	4-026276
CN20	R*4	4-026306	GAMA	R*4	4-026271	I	I*2	4-026167	IC000	I*2	4-026242
IP	I*2	4-026346	IPASS	I*2	4-026164	ATCH	I*2	4-026210	ISAT	I*2	4-026170
ISUM	I*2	4-026244	J	I*2	4-026276	K	I*2	4-026212	LATC	I*2	4-026174
LATM	I*2	4-026176	LOND	I*2	4-026202	LONDR	I*2	4-026200	LN0M	I*2	4-026204
KN03	R*4	4-026222	RN01	R*4	4-026226	RN02	R*4	4-026232	RN11	R*4	4-026236
SAS1Y	R*4	4-026342	SASX2	R*4	4-026332	SASY	R*4	4-026326	SASX	R*4	4-026322
SSR	R*4	4-026256	SUM	R*4	4-026252	VAR	R*4	4-026246	SASY2	R*4	4-026336
									SAS1	R*4	4-026316

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
C	R*4	4-026112	000030	12 (6)
FILNAM	L*1	4-000000	000036	15 (36)
ICLOUD	I*2	4-001062	001000	256 (16,16)
ID	L*1	4-000042	000012	5 (10)
IGAR	I*2	4-000062	001000	256 (16,16)
IGIN	I*2	4-015062	003000	768 (3,16,16)
IWATER	I*2	4-002062	001000	256 (16,16)
NAME	L*1	4-000054	000006	3 (6)
F	R*4	4-026062	000030	12 (6)
R	R*4	4-024062	002000	512 (16,16)
RE-ORT	L*1	4-000036	000004	2 (4)
VIN	R*4	4-003062	012000	2560 (5,16,16)
VINH	R*4	4-020062	004000	1024 (16,16,2)
X	R*4	4-026142	000020	8 (4)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000114	15	1-000136	25	**	30	**
42	1-001146	50	**	100	1-001506	150	**
178	1-002336	200	1-002476	400	**	500	1-004000
550	**	700	1-004474	1000	3-000000	1100	3-000042
1300	3-000076	1400	3-000156	1500	3-000356	1550	3-000474
1600	3-000720	1700	3-000762	1750	3-001012	1800	3-001006
						2000	3-000010

ORIGINAL PAGE IS  
OF POOR QUALITY

FORTRAN IV-PLUS V02-51E 14:33:33 30-OCT-80 PAGE 6  
POSTPP.FTN /TR:BLOCKS/WR

FUNCTIONS AND SUBROUTINES REFERENCED

OPEN

TOTAL SPACE ALLOCATED = 034214 7238

, TT17: [320, 40]A, B/-SP=SYO: [320, 40]P[3]TPP